UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

COMPUTATIONAL STUDY FOR THE NUMERICAL RESOLUTION OF THERMAL AND FLUID DYNAMIC PROBLEMS.

Studies: **MASTER'S DEGREE IN SPACE AND AERONAUTICAL ENGINEERING**

Student: **Cesar David Navas Prada**

Director: **Carlos David Pérez Segarra**

Codirector: **Asensio Oliva Llena**

Thesis Call: **QP 2017-2018**

Document: **Final Report**

Barcelona, September 2018

ABSTRACT

Since the development of the mechanical, aeronautical, and other industries along the computational technologies advances, a crucial field in engineering has studied the behavior and properties of fluids involved in engineering. During the last years this has led to advanced studies of the numerical resolution of Navier-Stokes equations, a relevant step to understand the fluid dynamics. Discretization methods, numerical schemes and solver methods has been developed; however, the application for specific cases and its effectivity or convenience for each case need to be thoroughly studied. Through the study of the convection-diffusion equation and the fractional step method and further implementation in thermal and fluid dynamic problems, this document analyzes the computational effects of these three components of CFD. Results shows how finer mesh sizes and discretization methods give a better performance in cases when the convective flows are higher than the diffusive flows. Contrarily, in relatively equal convective-diffusive flows the mesh size does not have a great influence, although finer meshes will have more precise results and better convergence features. Moreover, results show how numerical schemes can influence on the computational cost and time, especially for high-order schemes where stability is an important parameter to be aware. Finally, the analysis of solver methods demonstrates that the correct use and selection can be an important engineering decision because it could optimize the computational cost of the CFD studies applied to real fluid analysis.

I declare that,

The work in this Master Thesis is completely my own work,

No part of this Master Thesis is taken from other people's work without giving them credit,

All references have been cited,

I understand that an infringement of this declaration leaves me subject to the foreseen disciplinary actions by *The Universitat Politecnica de Catalunya – BarcelonaTECH.*

Cesar David Navas Prada		September 19, 2018
Student Name	Signature	Date

Title of the Thesis: COMPUTATIONAL STUDY FOR THE NUMERICAL RESOLUTION OF THERMAL AND FLUID DYNAMIC PROBLEMS.

Dedication:

To my mother. The person who has been an inspiration to me all this time, to the one who has always supported me and has taught me that life is a battle, but it is better when you have people backing you up.

To my father, little was the time with you, but it formed me the way I am now.

To my partner in life, she has enlightened my path even in those dark times.

Acknowledgements:

I would like to express my gratitude to the people who has been involved and helped me to achieve this academic goal.

First, to the director of my work, Carlos David Perez Segarra, and the CTTC (Centre Tecnologic de Transferencia de Calor) department of the Universitat Politecnica de Catalunya for giving me the opportunity to expand my knowledge on CFD and let me work this thesis with their supervision. Furthermore, the help I received from my director that was indispensable to finish my work.

Finally, I would like to thank to my professors during my master's degree and my classmates who supported me and gave me advice about my thesis work.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1. Aim of Study

The principal goal of this study is to perform an advanced computational study of the CFD (Computational Fluid Dynamics) components: discretization methods, numerical schemes and solver methods, by developing the CFD algorithms and validating the results with solved problems provided by the CTTC (Centre Tecnologic de Transferencia de Calor) from the Universitat Politecnica de Catalunya. Moreover, to correlates the physics and theoretical meaning applied to the numerical resolutions of the Navier-Stokes equations in thermal and fluid dynamic problems.

1.2. Scope of Study

The scope of this study is to provide a deeper and meaningful inside about the discretization, solver and numerical methods used on CFD. This study will require the development of codes to solve numerically the theoretical equations. It also will require the correct application and use of this method where the engineering concepts will be the principal tool to achieve the goal. Finally, the adequate interpretation of the results which will imply the combination of engineering concepts and physical interpretation. The three main scope areas of the study will be exposed as follows:

1.2.1. Code Development

A set of codes will be independently developed according to the specifications and needs of the CFD problems proposed. The codes required will be the followings:

- Navier-Stokes discretization (convection-diffusion equation, fractional step method)
- Numerical schemes (high and low order schemes)
- Solver methods (point-by-point and line-by-line)
- Turbulence study (Burgers equation)

1.2.2. Application Based on Engineering Concepts

Based on engineering concepts and CFD theory the adequate method and solution to each case, so then it could provide with some improvements for future works, determine and verify the convenience of some methods currently applied in CFD.

1.2.3. Interpretation and Discussion of Results

Three basic CFD areas will be covered:

- Evaluation of time and computational cost of implementing two different solver methods before mentioned
- Identification of weak and strong features of different numerical schemes, regarding accuracy, and computational cost as well.
- Analysis of the effects produced by the variation of physical parameters such as mesh sizes or Reynolds number.

1.3. Requirements

All work done, including reports, used codes and other resources must be personally develop and perform. Otherwise, any material of any kind used during the development of the project must be correctly referenced according to intellectual property laws. Lastly, a final report with achievements and considerations of the projects, and an annex document with all the extra material used and developed during the project will be handle for evaluation and consequently defended and exposed to a thesis jury. Among the two documents priory exposed, a budget document and a self-assessment report will be attached according to the Master´s thesis specifications (TFM) of the *Universitat Politécnica de Catalunya*.

1.4. Justification

Computational methods have become in recent years an important tool for engineers and a decisive area of optimization, research, and development in various engineering fields. Aeronautical, mechanical, structural and fluid engineering has focused this source to develop and thoroughly study the behaviors and physical parameter of fluids interacting with engineering devices and in nature as well. CFD (Computational Fluid Dynamics) has

been then a crucial tool for this development especially since the Aerospatiale and aeronautical industry started to increase. The computational methods implemented has now allowed engineers to determine and characterize the physic conditions of fluids such as pressure, temperature, and velocity. This study will be performed with the aim of improving designs and performance of engineering and give close-to-reality solutions to this complex physical phenomenon. For instance, Aerospatiale and aeronautic engineering has been highly benefited by CFD methods because of the study of the Navier-Stokes equations, which relates the principal parameters (velocity, pressure, temperature) necessary to design and build any flight machine which is exposed to the effect of aerodynamics and thermal exchange. Therefore, the study of Navier-stokes discretization and posterior resolution in this work will be the first relevant step for the analysis and engineering applications to aerodynamics and fluid systems constantly required in the mechanical, Aerospatiale, and aeronautical engineering.

1.5. State of the Art

The Navier-Stokes equations *Eq. 1* were developed in 1822 by Claude-Louis Navier and George Gabriel Stokes[1]. These equations in companion with the mass conservation equations *Eq. 2* gives the relation between pressure, temperature, density, and velocity fields on a fluid in motion. Since the appearance of the equations, the resolution of these equations has been widely studied in order to characterize the behavior of fluids in different conditions. However, the solution is not simple since for almost all real situations, the results are a combination of nonlinear partial differential equations. Computational Fluid dynamic (CFD) is the branch of engineering which has been studied the fluid's dynamics and heat transfer effects using algorithms, numerical techniques and discretization methods such as finite difference or finite elements to study fluid's behaviors under different scenarios and solve approximations to the equations in a computational environment using high speed processors.

$$\rho\left(\frac{\partial \vec{\upsilon}}{\partial t} + \vec{\upsilon}\cdot\nabla\vec{\upsilon}\right) = -\nabla p + \nabla\cdot T + \vec{f} \qquad \textit{(Eq. 1)}$$

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\vec{\upsilon}) = 0 \qquad \textit{(Eq. 2)}$$

CFD uses as a based the Navier-Stokes equations to determine the velocity and pressure fields in fluids, and for heat transfer effects it includes the energy equation which provides the temperature field to the solution. Nowadays, CFD has been mainly implemented through different software packages that facilitate the analysis in specific cases; for

instance, FLUENT and ANSYS-CFX[2]. However, some packages have a limited range of cases to be analyzed or cannot support cases with special boundary conditions or complicated geometries.

The core of CFD has been develop using computational resources and programing in basic computational languages, such as C++. Python, or MATLAB and through coding generating the numerical solutions faster and more accurate. The basic programming has been mainly focused in numerical solutions methods, discretization methods, and numerical schemes. Regarding numerical solutions methods, the accuracy and computational costs are the main characters to be studied. Point-by-point and Line-by-line methods are the easiest and simplest methods for numerical resolutions that the commercial software have been implemented. For the discretization methods, FVM (Finite Volume Method) has been the most used method because of its simplicity and computational advantages. Lastly, the numerical schemes can be classified into two categories: Low and high order. Low order schemes use less computational resources, but the accuracy tend to be low. The most used numerical schemes are the high-order schemes which provides a more accurate performance. However, high order schemes must be carefully implemented because of its instabilities in some cases depending of mesh sizes and Reynolds number.

An important phenomenon presented in fluid dynamics analysis that, CFD has been developing numerical solutions and analysis methods such as DNS, LES, or RANS, is the turbulence generated by high Reynolds numbers in fluid motions. Turbulence is one of the most studied fields and more difficult to analyze because it poses random changes of velocity in fluid flows.

Chapter 2

CONVECTION-DIFFUSION EQUATION

The governing equations of most of the properties of a fluid in motion can be expressed in a general formulation that is called the CONVECTION-DIFFUSION EQUATION *Eq. 3*. This equation pretends to generalize the convection and diffusion effects when a property is transported due to mass flows or particle diffuse effects on a fluid. Therefore, the convection-diffusion equation is a generalized formula which states that the accumulation of an arbitrary property through time plus the net convective flux in a specific volume is equal to the net diffusive flux plus a generation of the property per volume unit.

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \vec{\upsilon} \phi) = \nabla \cdot (\Gamma \nabla \phi) + S_{\phi} \qquad \textit{(Eq. 3)}$$

Where ϕ is the arbitrary property to be analyzed; for instance, velocity, Temperature, concentration, enthalpy, etc. ρ is the density, Γ is the diffusive coefficient which is specific and related to ϕ. Sϕ represents the generation of ϕ or external forces applied. Finally, υ is the flow velocity field. This flow field can be given by an external source trough an experiment or obtained analytically. Having defined the general equation, it is now easier and more efficient to analyze the governing equation of different fluid properties just by defining the convective, diffusive and source terms regarding the property analyzed. *Table 1* shows these terms considering each evaluated property to be replace in the convection-diffusion equation.

Equation	ϕ	Γ	S_{ϕ}
Continuity	1	0	0
Momentum	$\vec{\upsilon}$	μ	$-\nabla p + \nabla \cdot \vec{\tau} - \mu \nabla \vec{\upsilon} + \rho \vec{g}$
Energy	u	λ / Cv	$-\nabla \cdot \vec{g}^{R} + -p \nabla \cdot \vec{\upsilon} + \vec{\tau} : \nabla \vec{\upsilon}$
Specie k	Y_k	D_k	ω_k

Table 1. Convective, diffusive, and source terms to be replace in the convection-diffusion equation [3]

The convection-diffusion equation can even be rewritten assuming constant properties along the fluid domain (ρ and Γ) and assuming it is an incompressible flow. Therefore, the equation leads to:

$$\rho\frac{\partial\phi}{\partial t}+\rho(\vec{v}\cdot\nabla)\phi=\Gamma(\Delta\phi)+S_{\phi} \qquad \textit{(Eq. 4)}$$

During the development of this chapter the discretization of the convection-diffusion equation will be presented among with the computational requirements such as numerical schemes and solver methods. Finally, some cases of study will be developed for convection-diffusion effects and pure diffusion.

2.1. Discretization of the Convection-Diffusion Equation

There are three principal discretization methods to solve numerically these sorts of equations. Finite Difference Method (often used to solve partial differential equations), Finite Element Method (highly used in structural analysis), and Finite Volume Method. This last one FVM is the one chosen to be used in the discretization because it has more computational and convenient advantages over the other two methods. In order to use FVM, some parameters need to be defined. First, a structured two-dimensional (2D) mesh will be used. *Figure 1* shows the configuration of the mesh. The finite volume will be rectangular with a node-centered configuration. Hence, the volume will have equal dimensions along the vertical and horizontal axis. Furthermore, the distances between the main node and the neighbor nodes will again be the same along vertical and horizontal axis. The mesh will be analyzed on a Cartesian plane, so the horizontal axis will be known as X axis, and vertical as Y axis indeed.

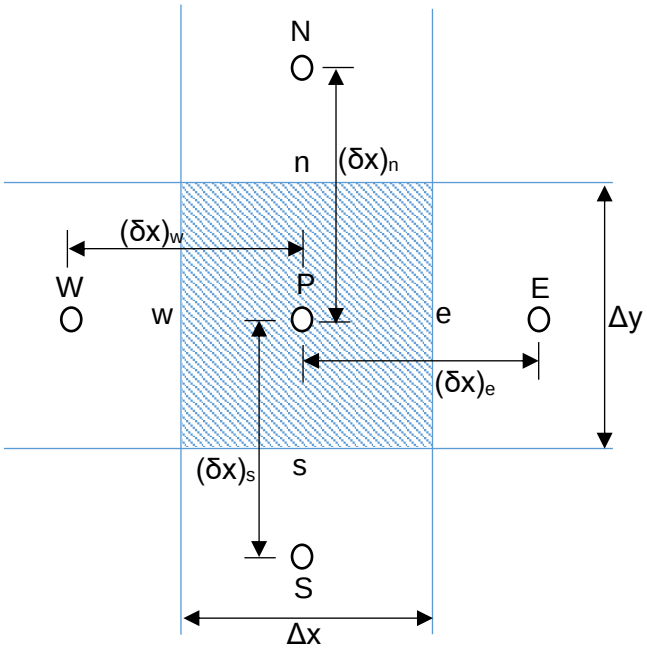


Figure 1. Scheme of the rectangular finite volume used in the discretization

It is possible now to discretize the convection-diffusion equation considering the following hypothesis[3]:

- The internal generation of the property represented by the source term (S_ϕ) is considered constant along the control volume and time.
- The control volumes, and distances along axis are considered constant during time, and Δt considered equal during each time step.
- Fluxes are constant in each face of the volume.
- Fully Implicit scheme (β = 1)
- Constant properties (ρ and Γ) trough time and along domain.
- A simpler notation *Figure 2* using $(\Omega)^n$ will represent the value of the property Ω at the previous instant, and Ω without a superscript is the value at the current time.

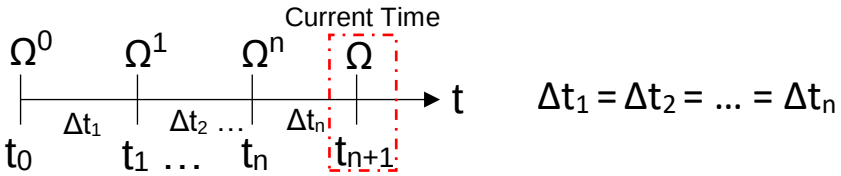


Figure 2. Time scheme and property notation

It is necessary, to discretize the partial differential diffusive and convective terms of the equation, to apply the Convergence Theorem *Eq. 5*. This theorem, also known as Gauss Theorem, correlates the divergence of a vector field in a closed surface with the fluxes on the outside of the surface. Therefore, the Convergence Theorem leads to:

$$\int_S (\nabla \cdot \vec{R})\, dA = \int_{\partial S} \vec{R} \cdot \hat{\boldsymbol{n}}\, ds \qquad \textit{(Eq. 5)}$$

Considering this hypothesis and the convergence theory, it is time now to discretize the general convection-diffusion equation *Eq. 3*.

First the time and volume integration are done and leads to *Eq. 6*.

$$\int_{t^n}^{t^{n+1}} \int_{Vp} \left(\frac{\partial \rho\phi}{\partial t} + \nabla \cdot (\rho \vec{v} \phi) \right) dV\, dt = \int_{t^n}^{t^{n+1}} \int_{Vp} \left(\nabla \cdot (\Gamma \nabla \phi) + S_\phi \right) dV\, dt \qquad \textit{(Eq. 6)}$$

The time dependent term or the accumulation of the property ϕ is discretized on the following approach:

$$\int_{t^n}^{t^{n+1}} \int_{Vp} \left(\frac{\partial \rho\phi}{\partial t} \right) dV\, dt \cong \int_{t^n}^{t^{n+1}} \left(\frac{\partial \rho\phi}{\partial t} \Delta x \Delta y \right) dt \cong \frac{(\rho\phi)_p - (\rho\phi)_p^n}{\Delta t} \Delta x \Delta y \qquad \textit{(Eq. 7)}$$

Then, by applying the convergence theory the convective and diffusive terms are discretized:

$$\int_{t^n}^{t^{n+1}} \int_{Vp} \left(\nabla \cdot (\rho \vec{v} \phi)\right) \, dV \, dt \cong \int_{dVp} (\rho \vec{v} \phi) \cdot \hat{\boldsymbol{n}} \, dS$$
$$\cong [(\rho u \phi)_e - (\rho u \phi)_w]\Delta y + [(\rho v \phi)_n - (\rho v \phi)_s]\Delta x \qquad \textit{(Eq. 8)}$$

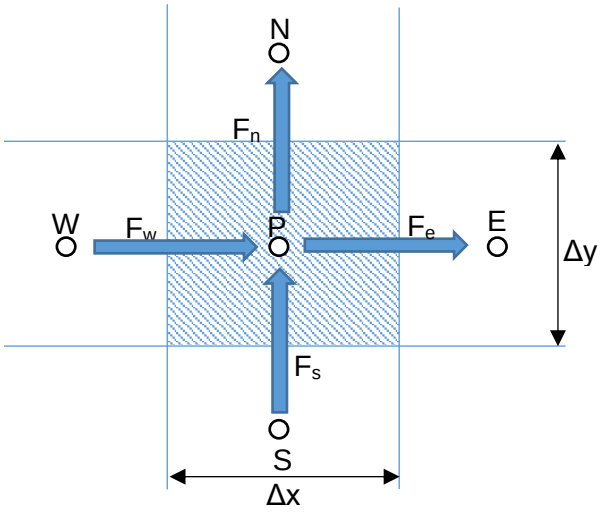


Figure 3. Fluxes scheme for the convective term

The sign of the convective fluxes through each face of the control volume are considered according to the convention stablished in *Figure 3*. Positive when the flow is exiting the control volume and negative when it is going inside. Furthermore, to analyze the velocity field, positive values will be in axis X when the flow goes from left to right (Eastward), and in axis Y when the flow goes bottom to top (Upward). Consequently, the diffusive term:

$$\int_{t^n}^{t^{n+1}} \int_{Vp} \left(\nabla \cdot (\Gamma \nabla \phi)\right) \, dV \, dt \cong \int_{dVp} (\Gamma \nabla \phi) \cdot \hat{\boldsymbol{n}} \, dS$$
$$\cong \left[\left(\Gamma \frac{\partial \phi}{\partial x}\right)_e - \left(\Gamma \frac{\partial \phi}{\partial x}\right)_w\right]\Delta y + \left[\left(\Gamma \frac{\partial \phi}{\partial y}\right)_n - \left(\Gamma \frac{\partial \phi}{\partial y}\right)_s\right]\Delta x \qquad \textit{(Eq. 9)}$$
$$\cong \left[\left(\Gamma \frac{\phi_e - \phi_P}{\delta x}\right) - \left(\Gamma \frac{\phi_P - \phi_w}{\delta x}\right)\right]\Delta y$$
$$+ \left[\left(\Gamma \frac{\phi_n - \phi_P}{\delta y}\right) - \left(\Gamma \frac{\phi_P - \phi_s}{\delta y}\right)\right]\Delta x$$

It is important to note that the velocity field vector on the convective term is decomposed into two components. The X component (horizontal) will be denoted from now one as **u** and the Y component (vertical) as **v.**

After that, the source term is integrated and discretized as well:

$$\int_{t^n}^{t^{n+1}} \int_{Vp} \left(S_\phi\right) \, dV \, dt \cong S_\phi \Delta x \Delta y \qquad \textit{(Eq. 10)}$$

Finally, Combining *Eq. 7, 8, 9, and 10* it is obtained:

$$\frac{(\rho\phi)_p - (\rho\phi)_p^n}{\Delta t}\Delta x\Delta y + [(\rho u\phi)_e - (\rho u\phi)_w]\Delta y + [(\rho v\phi)_n - (\rho v\phi)_s]\Delta x = \left[\left(\Gamma\frac{\phi_e - \phi_P}{\delta x}\right) - \left(\Gamma\frac{\phi_P - \phi_w}{\delta x}\right)\right]\Delta y + \left[\left(\Gamma\frac{\phi_n - \phi_P}{\delta y}\right) - \left(\Gamma\frac{\phi_P - \phi_s}{\delta y}\right)\right]\Delta x + S_\phi\Delta x\Delta y \quad \textit{(Eq. 11)}$$

Eq. 11 can be rewritten according to hypothesis and using a simpler notation:

$$\frac{\rho(\phi_p - \phi_p^n)}{\Delta t}\Delta x\Delta y + Fe\phi_e - Fw\phi_w + Fn\phi_n - Fs\phi_s = De(\phi_e - \phi_P) - Dw(\phi_P - \phi_w) + Dn(\phi_n - \phi_P) - Ds(\phi_P - \phi_s) + S_\phi\Delta x\Delta y \quad \textit{(Eq. 12)}$$

Where: $= \rho u_e\Delta y$; $Fw = \rho u_w\Delta y$; $Fn = \rho v_n\Delta x$; $Fs = \rho v_s\Delta x$

and $De = \frac{\Gamma_e\,\Delta y}{(\delta x)_{PE}}$; $Dw = \frac{\Gamma_w\,\Delta y}{(\delta x)_{PW}}$; $Dn = \frac{\Gamma_n\,\Delta x}{(\delta y)_{PN}}$; $De = \frac{\Gamma_s\,\Delta x}{(\delta y)_{PS}}$

It is indispensable now to discretize the continuity mass equation *Eq. 13*, this equation can be found using *Table 1*.

$$\int_{t^n}^{t^{n+1}}\int_{Vp}\left(\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\vec{v})\right)dV\,dt = 0 \quad \textit{(Eq. 13)}$$

Using the same approach that in prior discretization:

$$\frac{\rho_p - \rho_p^n}{\Delta t}\Delta x\Delta y + [(\rho u)_e - (\rho u)_w]\Delta y + [(\rho v)_n - (\rho v)_s]\Delta x = 0 \quad \textit{(Eq. 14)}$$

Using the same notation as before

$$\frac{\rho_p - \rho_p^n}{\Delta t}\Delta x\Delta y + Fe - Fw + Fn - Fs = 0 \quad \textit{(Eq. 15)}$$

Having define both equations, the continuity equation *Eq. 15* is substrate from the convection-diffusion equation *Eq. 12*.

$$\frac{\rho(\phi_p - \phi_p^n)}{\Delta t}\Delta x\Delta y + Fe(\phi_e - \phi_P) - Fw(\phi_w - \phi_P) + Fn(\phi_n - \phi_P) - Fs(\phi_s - \phi_P) = De(\phi_e - \phi_P) - Dw(\phi_P - \phi_w) + Dn(\phi_n - \phi_P) - Ds(\phi_P - \phi_s) + S_\phi\Delta x\Delta y \quad \textit{(Eq. 16)}$$

The source term can be treated as a linear function of ϕ. This term depends on the property, and therefore taking a linear variation will be easier and give more advantages that taking

it as a constant. If the source term does not have a linear form, it could be linearized by using *Eq. 17*[4].

$$S_\phi = S_C^\phi + S_P^\phi \phi_P \qquad \textit{(Eq. 17)}$$

Where S_C^ϕ is a constant, and S_P^ϕ is the rate of change regarding ϕ_P.

2.2. Numerical Schemes

Eq. 16 is the integrated and discretized form of the convection-diffusion equation considering a rectangular control volume and a 2D structured mesh. This equation is a function of dependent variables that further will be numerically resolve in Se*ction 3.3*. However, this equation is function of the property values at faces of the finite volume ($\phi_e, \phi_w, \phi_n, \phi_s$). This develops a problem when it comes to numerically solving a discretized equation because it will generate more variables that equations available because the solving process will be developed on each grid node of the mesh, not in each face. Therefore, the system of equations will not be able to be solved. Nonetheless, the values of these faces can be related to the values of the center grid node and its neighbors ($\phi_E, \phi_W, \phi_N, \phi_S, \phi_P$). The different strategies to find this value are called numerical schemes.

There are two main type of numerical schemes: Low-order and High-order numerical schemes. Low-order schemes approximate the value on the faces using the nearest neighbors' grid nodes in each direction ($\phi_E, \phi_W, \phi_N, \phi_S, \phi_P$). On the other hand, High-order schemes uses the first and second nearest grid nodes in each direction. ($\phi_E, \phi_{EE}, \phi_W, \phi_{WW}, \phi_N, \phi_{NN}, \phi_S, \phi_{SS}, \phi_P$).

2.2.1. Low-order Numerical Schemes

As priory explained, a low-order scheme relates the main node with its neighbors to find the property values at each face. Hence, it leads to:

$$\phi_e = f(\phi_E, \phi_P)\ ;\ \phi_w = f(\phi_W, \phi_P)\ ;\ \phi_n = f(\phi_N, \phi_P)\ ;\ \phi_s = f(\phi_S, \phi_P) \qquad \textit{(Eq. 18)}$$

By using the finite volume scheme utilized *Figure 1* the value that ϕ takes at each face depends mainly on the low-order scheme selected. There are five most relevant schemes that will be discussed in this document[4]:

- **Upwind Scheme (UDS):** This scheme formulates just the convective terms, leaving the diffusive one intact. UDS uses the flow direction at the control volume and define

a correlation between the convective fluxes and the grid nodes values of ϕ. This scheme offers a more reasonable physical view of the convective effect. This scheme states that the value of ϕ at each face depends on the convective flux that is crossing the face, and it is equal to the value of ϕ at the upwind grid node. Therefore, UDS can be formulated with the following set of equations:

$$\phi_e = \begin{cases} \phi_P \; if \; Fe > 0 \\ \phi_E \; if \; Fe < 0 \end{cases} \quad \textit{(Eq. 19)}$$

$$\phi_w = \begin{cases} \phi_W \; if \; Fw > 0 \\ \phi_P \; if \; Fe < 0 \end{cases} \quad \textit{(Eq. 20)}$$

$$\phi_n = \begin{cases} \phi_P \; if \; Fn > 0 \\ \phi_N \; if \; Fn < 0 \end{cases} \quad \textit{(Eq. 21)}$$

$$\phi_s = \begin{cases} \phi_S \; if \; Fe > 0 \\ \phi_P \; if \; Fe < 0 \end{cases} \quad \textit{(Eq. 22)}$$

- **Central Difference Scheme (CDS):** this scheme evaluates the value of ϕ using the arithmetic mean of the values at both neighbor grid nodes. This scheme was the first scheme to be assumed to evaluate the convective terms. Although it is simple to apply, it is not much accurate and not highly used. This scheme leads the next equations:

$$\phi_e = \frac{1}{2}(\phi_P + \phi_E) \quad \textit{(Eq. 23)}$$

$$\phi_w = \frac{1}{2}(\phi_P + \phi_W) \quad \textit{(Eq. 24)}$$

$$\phi_n = \frac{1}{2}(\phi_P + \phi_N) \quad \textit{(Eq. 25)}$$

$$\phi_s = \frac{1}{2}(\phi_P + \phi_S) \quad \textit{(Eq. 26)}$$

- **Exponential Scheme (EDS):** this scheme finds the exact solution of the convection-diffusion equation for a one-dimensional 1D case not regarding source term and time-dependency**.** It considers the total convective and diffusive fluxes a face and by derivation stablish that:

$$J = \rho u \phi - \Gamma \frac{d\phi}{dx} \quad \textit{(Eq. 27)}$$

$$\frac{dJ}{dx} = J_e - J_w = 0 \quad \textit{(Eq. 28)}$$

$$\therefore Fe\left(\phi_P + \frac{\phi_P - \phi_E}{e^{Pe} - 1}\right) - Fw\left(\phi_W + \frac{\phi_W - \phi_P}{e^{Pw} - 1}\right) = 0 \quad \textit{(Eq. 29)}$$

Where $Pe = \frac{Fe}{De}$ and it is the PECLET number. it will be defined in *Chapter 2.3*.

- **The Hybrid Scheme (HDS):** this scheme is a combination of the upwind and the central difference schemes. For low convective fluxes this scheme uses the CDS (-2 < Pe< 2), and for ranges out of that it uses the UDS. However, in the UDS region the diffusive flux is set to zero. Hence, this scheme is highly unstable for high convective fluxes.
- **Power-Law Scheme (PLDS):** this scheme linearized in a fifth power the curve of the exact solution given by EDS and find the value at each face with this formula. The solution given by this scheme is highly accurate, but the computational cost is directly proportional. Both, EDS and PLDS, are the most computational expensive methods to apply.

2.2.2. High-order Numerical Schemes

High-resolution schemes implies the inclusion of more than one neighbor grid point in each direction to find the value of ϕ at each face[5]. The scheme that will be discussed are based on two neighbor point for each direction. Therefore, the scheme of the finite volume for HRS is as shown in *Figure 4*. With HRS some numerical errors that are present on first and second order schemes (low-order schemes) are avoided. This means that the accuracy of these schemes is improved by considering more grid points; hence, the value at the central point is influenced by more precise values of the surroundings. HRS generates more variables on the general equation *Eq. 13*. First, because now the values at each face have one more grid point to analyze. *Eq. 18* then needs to be included the value at the second neighbor grid point.

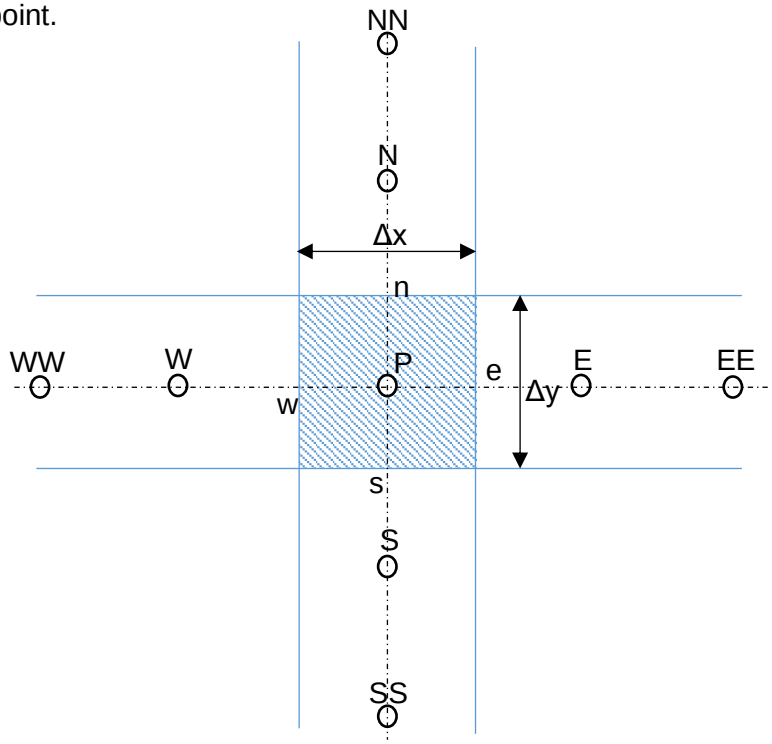


Figure 4. Scheme of the finite volume for high-order schemes

$$\phi_e = f(\phi_E, \phi_{EE}, \phi_W, \phi_P)\,;\ \phi_w = f(\phi_W, \phi_{WW}, \phi_E, \phi_P)\,;$$
$$\phi_n = f(\phi_N, \phi_{NN}, \phi_S, \phi_P)\,;\quad \phi_s = f(\phi_S, \phi_{SS}, \phi_N, \phi_P) \qquad \textit{(Eq. 30)}$$

Furthermore, the discretized equation *Eq. 16* needs to be modified to be adapted to the high-order schemes. The deferred correction method which was used on this document rewrite the convective fluxes at each face with the following formula *Eq. 31*[5]:

$$J_f = F_f\phi_f = F_f\phi_f^U - F_f(\phi_f^U - \phi_f^{HRS}) \qquad \textit{(Eq. 31)}$$

Where subscript f means the values at the face, and the superscripts U and HRS denotes respectively the values of ϕ using upwind scheme and HRS schemes.

Integrating *Eq. 31* on *Eq. 16* we obtain then:

$$\begin{aligned}\frac{\rho(\phi_p - \phi_p^n)}{\Delta t}\Delta x\Delta y &+ Fe(\phi_e^U - \phi_P) - Fw(\phi_w^U - \phi_P) + Fn(\phi_n^U - \phi_P) \\ &- Fs(\phi_s^U - \phi_P) \\ &= De(\phi_e - \phi_P) - Dw(\phi_P - \phi_w) + Dn(\phi_n - \phi_P) \\ &- Ds(\phi_P - \phi_s) - Fe(\phi_e^{HRS} - \phi_e^U) + Fw(\phi_w^{HRS} - \phi_w^U) \\ &- Fn(\phi_n^{HRS} - \phi_n^U) + Fs(\phi_s^{HRS} - \phi_s^U) + S_\phi\Delta x\Delta y\end{aligned} \qquad \textit{(Eq. 32)}$$

Where ϕ_f^U can be calculated using *Eqs. 19,20,21,22*.

In order to find ϕ_f^{HRS} it will depend on the HRS chosen. Nonetheless, the method of the normalized variables is applied to make simpler the process and generalized the schemes equations into one polynomial equation that could be numerically solved. It should be highlighted that the normalized variables method is often used for High-order schemes and for non-structured meshes. However, in this work it will be used because it makes easier the process of adding new schemes, the code is general and simple, and it could be used for uniform and no-uniform meshes.

2.2.2.1. Normalized Variables Notation

This method derives the convective flux using two grid points upstream and one downstream the flux going through each control volume face[5]. Thus, in total 3 grid points per each face will be considered in the equation. *Figure 5* shows the notation used.

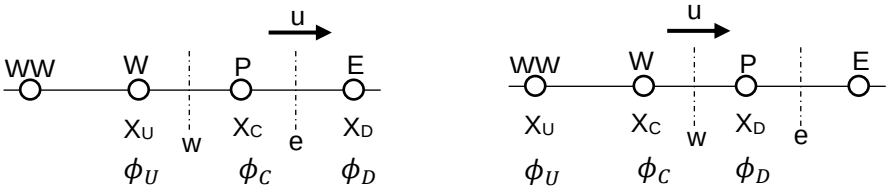


Figure 5. Normalized variables scheme and notation

The normalized variables then are found using the following formulas:

$$\bar{\phi} = \frac{\phi - \phi_U}{\phi_D - \phi_U} \quad ; \quad \bar{x} = \frac{x - x_U}{x_D - x_U} \qquad \textit{(Eq. 33)}$$

The most common high-order schemes used this notation and they will be presented on the next *Table 2.*

HRS method	Normalized value at face $\bar{\phi}_f$	
Second-order Upwind	$\bar{\phi}_f = \frac{\bar{x}_f}{\bar{x}_C}\bar{\phi}_C$	*(Eq. 34)*
QUICK	$\bar{\phi}_f = \bar{x}_f + \frac{\bar{x}_f(\bar{x}_f - 1)}{\bar{x}_C(\bar{x}_C - 1)}(\bar{\phi}_C - \bar{x}_C)$	*(Eq. 35)*
Fromm	$\bar{\phi}_f = \bar{\phi}_C + (\bar{x}_f - \bar{x}_C)$	*(Eq. 36)*
SMART	$if\ 0 < \bar{\phi}_C < \frac{\bar{x}_f}{3} \implies \bar{\phi}_f = -\frac{\bar{x}_f(1 - 3\bar{x}_C + 2\bar{x}_f)}{\bar{x}_C(\bar{x}_C - 1)}\bar{\phi}_C$ $if\ \frac{\bar{x}_f}{3} < \bar{\phi}_C < \frac{\bar{x}_C}{\bar{x}_f}(1 - \bar{x}_C + \bar{x}_f) \implies QUICK\ Eq.34$ $if\ \frac{\bar{x}_C}{\bar{x}_f}(1 - \bar{x}_C + \bar{x}_f) < \bar{\phi}_C < 1 \implies \bar{\phi}_f = 1$ $elsewhere \implies \bar{\phi}_f = \bar{\phi}_C$	*(Eq. 37)*

Table 2. Normalized values at face for high-order schemes[5]

2.3. Numerical Solver Methods

After the discretization and definition of the convection-diffusion equation, it is time to solve it numerically by a general algebraic equation applied to each grid point of the mesh. Until now, it is known from the general *Eq. 32* that the value of ϕ at the central node P will be a function of the values at the faces of the control volume.

$$\phi_P = f(\phi_e^{HRS}, \phi_e^U, \phi_w^{HRS}, \phi_w^U, \phi_n^{HRS}, \phi_n^U, \phi_s^{HRS}, \phi_s^U) \qquad \textit{(Eq. 38)}$$

However, the U and HRS values at each face is function of the neighbor's grid points.

$$\phi_e^U = f(\phi_E, \phi_P)\ ;\ \phi_w^U = f(\phi_W, \phi_P)\ ;\ \phi_n^U = f(\phi_N, \phi_P)\ ;\ \phi_s^U = f(\phi_S, \phi_P) \qquad \textit{(Eq. 39)}$$

$$\phi_e^{HRS} = f(\phi_E, \phi_{EE}, \phi_W, \phi_P)\ ;\ \phi_w^{HRS} = f(\phi_W, \phi_{WW}, \phi_E, \phi_P)\ ;$$
$$\phi_n^{HRS} = f(\phi_N, \phi_{NN}, \phi_S, \phi_P)\ ;\ \phi_s^{HRS} = f(\phi_S, \phi_{SS}, \phi_N, \phi_P) \qquad \textit{(Eq. 40)}$$

Therefore, *Eq. 38* is rewritten:

$$\phi_P = f(\phi_E, \phi_{EE}, \phi_W, \phi_{WW}, \phi_N, \phi_{NN}, \phi_S, \phi_{SS}) \qquad \textit{(Eq. 41)}$$

The algebraic equation to solve the value of ϕ_P needs to be general, either for low-order or high-order schemes. Low-order schemes just depend on the first neighbor grid points, so the general algebraic equation will be a function of these principal nodes ($\phi_E, \phi_W, \phi_N, \phi_S$), and for the high-order schemes an extra term in function of the second neighbor grid points will be added ($\phi_{EE}, \phi_{WW}, \phi_{NN}, \phi_{SS}$).

The general algebraic equation is given by *Eq. 42*:

$$a_P\phi_P = a_E\phi_E + a_W\phi_W + a_N\phi_N + a_S\phi_S + b + b_{dc} \qquad \textit{(Eq. 42)}$$

Where the coefficients $a_P, a_E, a_W, a_N, a_S, b, b_{dc}$ can be found using *Tables 3 and 4* depending if a high-order or low-order scheme is being applied. It should be emphasized that the term b_{dc} is the results of applying the deferred correction method applied priory on *Chapter 2.2.2* for high-order schemes. Therefore, this term is just considered to solve *Eq. 42* when HRS is being used. Otherwise, this term is considered zero for low-order schemes.

The values with superscript HRS in *Eq. 49* are calculated using *Table 2* regarding the scheme used, and the values with superscript U are found with *Eq. 19,20,21,22*.

Coefficient	Value for High-order schemes	
a_E	$a_E = De + \max(-Fe, 0) \quad ; \quad De = \frac{\Gamma_e\,\Delta y}{(\delta x)_{PE}} \quad Fe = \rho u_e \Delta y$	*(Eq. 43)*
a_W	$a_W = Dw + \max(Fw, 0) \; ; \; Dw = \frac{\Gamma_w\,\Delta y}{(\delta x)_{PW}} \quad Fw = \rho u_w \Delta y$	*(Eq. 44)*
a_N	$a_N = Dn + \max(-Fn, 0) \quad ; \quad Dn = \frac{\Gamma_n\,\Delta x}{(\delta y)_{PN}} \quad Fn = \rho v_n \Delta x$	*(Eq. 45)*
a_S	$a_S = Ds + \max(Fs, 0) \quad ; \quad Ds = \frac{\Gamma_s\,\Delta x}{(\delta y)_{PS}} \quad Fs = \rho v_s \Delta x$	*(Eq. 46)*
a_P	$a_P = a_E + a_W + a_N + a_S + \frac{\rho(\Delta x \Delta y)}{\Delta t}$	*(Eq. 47)*
b	$b = \frac{\rho(\Delta x \Delta y)}{\Delta t}\phi_p^n + (S_C^\phi + S_P^\phi \phi_P)\Delta x \Delta y$	*(Eq. 48)*
b_{dc}	$b_{dc} = -Fe(\phi_e^{HRS} - \phi_e^U) + Fw(\phi_w^{HRS} - \phi_w^U) - Fn(\phi_n^{HRS} - \phi_n^U) + Fs(\phi_s^{HRS} - \phi_s^U)$	*(Eq. 49)*

Table 3. Coefficients for the algebraic equation using high-order schemes

Coefficient	Value for Low-order schemes	
a_E	$a_E = De \cdot \mathrm{A}(\lvert \mathrm{P}_e \rvert) + \max(-Fe, 0)\,;\; De = \frac{\Gamma_e\,\Delta \mathrm{y}}{(\delta x)_{PE}}\;\; Fe = \rho \mathrm{u}_e \Delta y$	*(Eq. 50)*
a_W	$a_W = Dw \cdot \mathrm{A}(\lvert \mathrm{P}_w \rvert) + \max(Fw, 0)\,; Dw = \frac{\Gamma_w\,\Delta \mathrm{y}}{(\delta x)_{PW}}\;\; Fw = \rho \mathrm{u}_w \Delta y$	*(Eq. 51)*
a_N	$a_N = Dn \cdot \mathrm{A}(\lvert \mathrm{P}_n \rvert) + \max(-Fn, 0)\,;\; Dn = \frac{\Gamma_n\,\Delta \mathrm{x}}{(\delta y)_{PN}}\;\; Fn = \rho \mathrm{v}_n \Delta x$	*(Eq. 52)*
a_S	$a_S = Ds \cdot \mathrm{A}(\lvert \mathrm{P}_s \rvert) + \max(Fs, 0)\;\;;\;\; Ds = \frac{\Gamma_s\,\Delta \mathrm{x}}{(\delta y)_{PS}}\;\; Fs = \rho \mathrm{v}_s \Delta x$	*(Eq. 53)*
a_P	$a_P = a_E + a_W + a_N + a_S + \frac{\rho(\Delta x \Delta y)}{\Delta t}$	*(Eq. 54)*
b	$b = \frac{\rho(\Delta x \Delta y)}{\Delta t}\phi_p^n + (\mathrm{S}_C^{\phi} + \mathrm{S}_P^{\phi}\phi_P)\Delta x \Delta y$	*(Eq. 55)*
b_{dc}	$b_{dc} = 0$	*(Eq. 56)*

Table 4. Coefficients for the algebraic equation using low-order schemes

Where the function $\mathrm{A}(\lvert \mathrm{P}_f \rvert)$ depends on the PECLET number and it is defined depending on the low-order scheme that will be used (see *Table 5*). The Peclet number P is the ratio between the convective and the diffusive fluxes on each face. Thus, this number determines the proportion of convective or diffusive fluxes on the faces of the control volume.

$$\mathrm{P}_e = \frac{Fe}{De} \;\;;\;\; \mathrm{P}_w = \frac{Fw}{Dw} \;\;;\;\; \mathrm{P}_n = \frac{Fn}{Dn} \;\;;\;\; \mathrm{P}_s = \frac{Fs}{Ds} \qquad \textit{(Eq. 57)}$$

Scheme	Value of $\mathrm{A}(\lvert \mathrm{P}_f \rvert)$
Upwind Scheme	1
Central Difference Scheme	$1 - 0.5\lvert \mathrm{P}_f \rvert$
Exponential Scheme	$\frac{\lvert \mathrm{P}_f \rvert}{e^{\lvert \mathrm{P}_f \rvert} - 1}$
Hybrid Scheme	$\max(0, 1 - 0.5\lvert \mathrm{P}_f \rvert)$
Power-law Scheme	$\max(0, (1 - 0.1\lvert \mathrm{P}_f \rvert)^5)$

Table 5. Function A(|Pf|) for low-order schemes[4]

Peclet numbers that tends to zero $\mathrm{P} \approx 0$ means that the diffusive flux is highly greater than the convective flux on the analyzed face, or so called pure-diffusion problem (conduction problem). On the other hand, Peclet numbers that tend infinity or have great values $\mathrm{P} \approx \infty$ shows that the convective flux is much greater than the diffusive flux, or pure-convective

problems[4]. On the same analysis $\mathrm{P} \approx 1$ represents that both convective and diffusion effects are produced equally on the face evaluated.

The next procedure is to solve the system of equations with the form of *Eq. 42* and the dimension of the matrix will be given by the number of grid points imposed on the creation of the mesh. Finer meshes will generate more grid points; therefore, more equations.

There are two main solver methods that could be implemented. The simplest method is the Gauss-Seidel method, or so-called Point-by-point. An enhanced method that uses TDMA (Tri-diagonal matrix algorithm) is the second option. Choosing an adequate solver method is a decisive step on the numerical solution.

2.3.1. Point-by-point Method

One of the simplest method to solve the system of equations o the point-by-point method, or Gauss-Seidel method[4]. As its name states, this method requires visiting each grid point of the mesh and calculate the value ϕ. This method requires iteration up until the solution converges. *Eq. 42* can be rewritten in order to be used for 2D or 3D studies as:

$$a_P \phi_P = a_{nb} \phi_{nb} + b \qquad \textit{(Eq. 58)}$$

Where the subscript nb means neighbors. Thus, the value ϕ_P depends of its neighbor's values and b and can be found by:

$$\phi_P = \frac{a_{nb} \phi_{nb}^* + b}{a_P} \qquad \textit{(Eq. 59)}$$

ϕ_{nb}^* is the value of ϕ in the neighbors grid points on the prior iteration or the guessed value. It should be noted that this process is repeated for each node of the mesh. Therefore, the more nodes on the mesh, the more time and computational cost will be necessary. This process requires then the storage of the values calculated before, and in every iteration these values are rewritten, so then these values can be used for the next iteration and continue up until the convergence desired (maximum error imposed) is reached.

One of the principal disadvantages of this method is the fact that it convergence time is quite large compared with other methods. Moreover, this method does not always converge. Nonetheless, a criterion could be applied to Gauss-Seidel method to determine its convergence. This criterion was posed by Scarborough in 1958 and it states that the convergence is guaranteed if the equation satisfies that[4]:

$$\frac{\sum|a_{nb}|}{|a_P|}\begin{cases}\leq 1 & for\ all\ equations \\ < 1 & for\ at\ least\ one\ equation\end{cases} \qquad \textit{(Eq. 60)}$$

2.3.2. Line-by-line Method

An enhanced method that uses TDMA and Gauss-Seidel method as combination can improved the convergence issue with the point-by-point Gauss-Seidel solver approach[4]. This method solves consecutive grid nodes on a same direction, either horizontal (X) or vertical (Y), by assuming the value of ϕ on the parallel line of grid nodes. Hence, each solving line becomes a 1D problem. While solving each line, a TDMA is generated, so the solver generates as many TDMA's as lines evaluated on the mesh. To clarify the concept *Figure 6,* represent the process in a structured mesh with [NxM] grid points. The vertical grid line composed by the grid nodes in position i (represented with bold dots) is the line chosen to be solved. If the values of ϕ at the neighbor lines $i-1$ and $i+1$ (represented with bold crosses) are considered known, then it could be considered a 1D problem with M grid points on the vertical axis, and TDMA method can be easily applied. The values at the neighbor lines (Bold crosses) are the values priory found and that are storage.

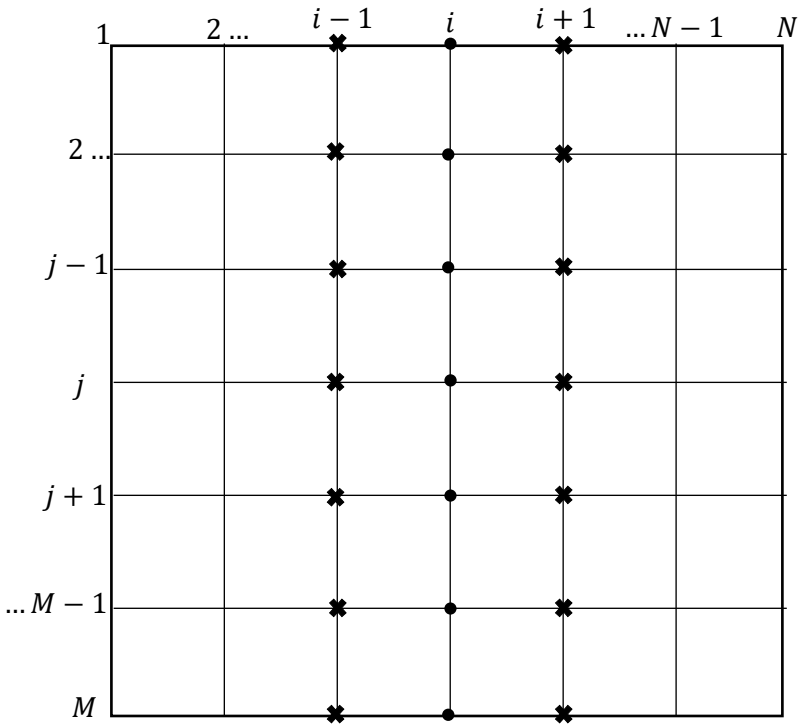


Figure 6. Graphical scheme for line-by-line solver method

This process is repeated N times, sweeping the vertical lines from left to right, or from right to left. Nonetheless, this process could be implemented on both directions. *Figure 6* used the vertical grid lines (columns), but the process can be implemented reversely for horizontal lines (rows). If desired, it could be applied in both direction, one after another, to obtain a more accurate solution. This method converges faster because the information or values from the boundaries are brought directly to the middle grid nodes. In some cases, it is important to choose the boundary line with defined values to start the "sweeping".

Considering now that the problem is 1D, it is possible to conclude that:

$$a_i\phi_i = a_{i-1}\phi_{i-1} + a_{i+1}\phi_{i+1} + b_i \qquad \textit{(Eq. 61)}$$

Note: *Eq. 61* is using subscript notation i for horizontal grid lines, it is just a generalized notation, for the example of *Figure 6* it could be replace for subscript j and have the same process.

If the boundary condition ϕ_1 is completely known, and according to *Eq. 61* ϕ_2 is known in terms of ϕ_1 and ϕ_3, then ϕ_2 becomes merely a function of ϕ_3. Consequently the same process for ϕ_3 where it is a function of the next value ϕ_4, Continuing until ϕ_{N-1} is a function of ϕ_N where it is the other boundary value[4]. Therefore, it is obtained the following form of equation:

$$\phi_i = P_i\phi_{i+1} + Q_i \qquad \textit{(Eq. 62)}$$

By following the same idea, we find that:

$$\phi_{i-1} = P_{i-1}\phi_i + Q_{i-1} \qquad \textit{(Eq. 63)}$$

Then combining Eq. 63 and Eq. 61 the coefficients P and Q are found:

$$P_i = \frac{a_{i-1}}{a_i - a_{i+1}P_{i-1}} \qquad \textit{(Eq. 64)}$$

$$Q_i = \frac{b_i + a_{i+1}Q_{i-1}}{a_i - a_{i+1}P_{i-1}} \qquad \textit{(Eq. 65)}$$

The values of the coefficients regarding the direction chosen for the line-by-line method can be found using the equations in *Table 6**.

Direction	P_i	Q_i	b_i	ϕ_P
X (rows)	$\frac{a_E}{a_P - a_W P_{i-1}}$	$\frac{b_P + a_W Q_{i-1}}{a_P - a_W P_{i-1}}$	$a_N\phi_N^n + a_S\phi_S^n + b + b_{dc}$	$P_i\phi_E^n + Q_i$
Y (columns)	$\frac{a_S}{a_P - a_N P_{j-1}}$	$\frac{b_P + a_N Q_{j-1}}{a_P - a_N P_{j-1}}$	$a_E\phi_E^n + a_W\phi_W^n + b + b_{dc}$	$P_j\phi_S^n + Q_j$

*Table 6. Coefficients for line-by-line method according to direction**

*The values with superscript $\phi_E^n, \phi_W^n, \phi_N^n, \phi_S^n$ are the values priory storaged, and the "sweeping" process is from left to right and from up to down. If the sweeping direction is inverted, then replace the coefficients:

$(a_E \Leftrightarrow a_W\ ;\ a_N \Leftrightarrow a_S\ ;\ Q_{i-1} \Rightarrow Q_{i+1}\ ;\ P_{i-1} \Rightarrow P_{i+1}\ ;\ \phi_E^n \Rightarrow \phi_W^n\ ;\ \phi_S^n \Rightarrow \phi_N^n)$.

2.4. Cases of Study: Convection-Diffusion Equation

2.4.1. Diagonal Flow Case

2.4.1.1. Objective

Once the convection-diffusion equation is discretized, the first case of study can be applied to verify the effectivity of the code generated. This problem is proposed by the CTTC and it consist of the transportation of a variable in a diagonal constant velocity field in steady state. The control volume is a square and both low-order and high-order schemes will be tested.

2.4.1.2. Problem Definition

This case considers a diagonal velocity field on 2D control volume with equal dimension in X and Y. Therefore, the flow is at 45° on the main diagonal of the square. The scheme and the boundary conditions are shown in *Figure 7.*

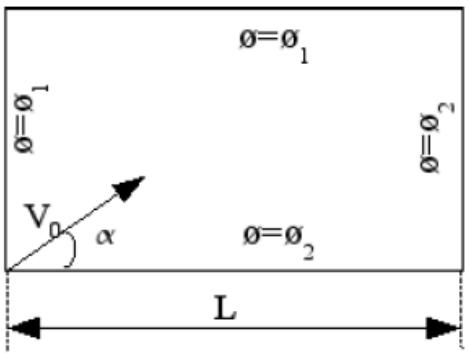


Figure 7. Diagonal Flow: Scheme and boundary conditions[6]

The velocity field then is given by *Eq. 66*.

$$u(x,y) = V_0 \cdot \cos(\alpha)$$
$$v(x,y) = V_0 \cdot \sin(\alpha) \qquad \textit{(Eq. 66)}$$

Where $\alpha = 45°$

The solution for an infinite Peclet number will have the form:

$\phi = \phi_1 \; above \; diagonal \; ; \; \phi = \phi_2 \; below \; diagonal$

In order to gain insights on the convective and diffusive effects, three different values of ρ/Γ will be used: $10, 10^3, 10^6$.

2.4.1.3. Discretization

The general convection-diffusion equation is given in *Eq. 4*. It should be recall that one of the assumptions made to find this equation was that the physical properties were constant in all domain. The same assumption will be made here; thus, rearranging this equation to have the coefficient required of ρ/Γ it will be easier to evaluate the different values on the equation. Moreover, regarding the fact that the source term is considered zero for this problem.

$$\frac{\rho}{\Gamma}\frac{\partial \phi}{\partial t} + \frac{\rho}{\Gamma}(\vec{v} \cdot \nabla)\phi = (\Delta \phi) \qquad \textit{(Eq. 67)}$$

Then, by using the same discretization approach that was used in *Chapter 2.3* and a 2D structured mesh (*Figure 1* and *Figure 8*) *Eq. 32* is defined:

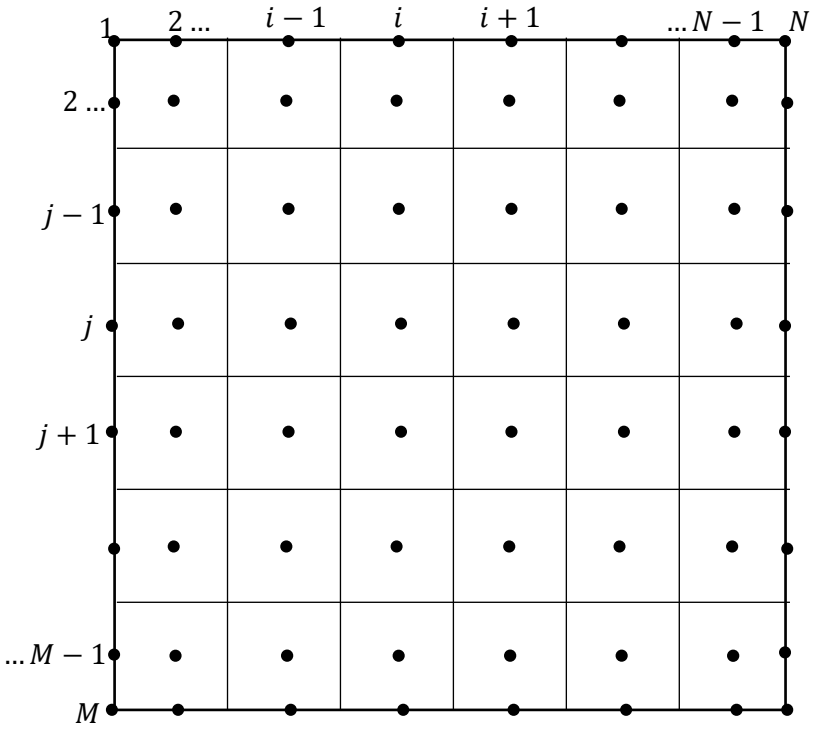


Figure 8. Diagonal Flow: 2D structured mesh scheme

It is important to note first that the boundary grid nodes are conveniently collocated on the wall faces of the total domain *Figure 8*. This allows the direct determination of boundary conditions, but the distance between these nodes and its neighbors is half of the central grid nodes. Secondly, that this configuration is valid for both, low-order and high-order schemes because the desire of comparison between them. The general solving equation (*Eq. 42*) could be applied to solve numerically the problem; consequently, the coefficients for boundary grid nodes *Table 7*, and middle grid nodes *Table 8* are found as follows:

Coefficient	Values of boundary conditions			
	West Nodes	East Nodes	South Nodes	North Nodes
a_E	0	0	0	0
a_W	0	1	0	0
a_N	0	0	0	0
a_S	0	0	0	1
a_P	1	1	1	1
b	1	0	0	0
b_{dc}	0	0	0	0

Table 7. Diagonal Flow: Coefficients for boundary conditions

Coefficient	Values for Middle Nodes	
a_E	$a_E = De + \max(-Fe, 0) \quad ; \quad De = \frac{\Delta y}{(\delta x)_{PE}} \quad Fe = \frac{\rho}{\Gamma} u_e \Delta y$	*(Eq. 68)*
a_W	$a_W = Dw + \max(Fw, 0) \; ; \; Dw = \frac{\Delta y}{(\delta x)_{PW}} \quad Fw = \frac{\rho}{\Gamma} u_w \Delta y$	*(Eq. 69)*
a_N	$a_N = Dn + \max(-Fn, 0) \quad ; \quad Dn = \frac{\Delta x}{(\delta y)_{PN}} \quad Fn = \frac{\rho}{\Gamma} v_n \Delta x$	*(Eq. 70)*
a_S	$a_S = Ds + \max(Fs, 0) \quad ; \quad Ds = \frac{\Delta x}{(\delta y)_{PS}} \quad Fs = \frac{\rho}{\Gamma} v_s \Delta x$	*(Eq. 71)*
a_P	$a_P = a_E + a_W + a_N + a_S + \frac{\rho}{\Gamma} \frac{(\Delta x \Delta y)}{\Delta t}$	*(Eq. 72)*
b	$b = \frac{\rho}{\Gamma} \frac{(\Delta x \Delta y)}{\Delta t} \phi_p^n$	*(Eq. 73)*
b_{dc}	$b_{dc} = -Fe(\phi_e^{HRS} - \phi_e^U) + Fw(\phi_w^{HRS} - \phi_w^U) - Fn(\phi_n^{HRS} - \phi_n^U) + Fs(\phi_s^{HRS} - \phi_s^U)$	*(Eq. 74)*

Table 8. Diagonal Flow: Coefficients for middle nodes

It should be noted that the values of ϕ_1 and ϕ_2 were assumed as 1 and 0 respectively. And the value of $V_0 = 1$. Furthermore, the values at the corners are assumed to be the arithmetic average of its neighbors. Therefore, the coefficients for the four corner grid points are shown in *Table 9*.

Coefficient	Values of corners grid points			
	West - North	East - North	West - South	East - South
a_E	0.5	0	0.5	0
a_W	0	0.5	0	0.5
a_N	0	0	0.5	0.5
a_S	0.5	0.5	0	0
a_P	1	1	1	1
b	0	0	0	0
b_{dc}	0	0	0	0

Table 9. Diagonal Flow: Coefficients for the corner grid points

Two important comments should be highlighted about the solution of this case. First, even though the problem asks for the steady condition, the terms that contain time dependency were considered because this study will analyze the time and computational cost for different numerical schemes and solver methods. This leads to the second consideration, and it is the fact that the case was solved using line-by-line and point-by-point method explained in *Chapter 2.3*, in the same way, three schemes were used: Upwind, second-order upwind and QUICK explained in *Chapter 2.2*. (Find values $\phi_e^{HRS}, \phi_w^{HRS}, \phi_n^{HRS}, \phi_s^{HRS}$ in *Table 2*)

2.4.1.4. Algorithm

With the purpose of graphically explain the code done for this case, *Figure 9* shows the algorithm flow chart that briefly explain the main processes inside the code. The code consists of one main code, which is the one that is used to manually enter parameters and run the code, and three more sub-codes. The mesh construction is one of those codes. The computation of discretization coefficients and finally the sub-code used to solve the system of equations. *Figure 9* is implemented three times, each time for a different value of ρ/Γ specified before. See *Attachment I – 2.1* for the algorithm implemented

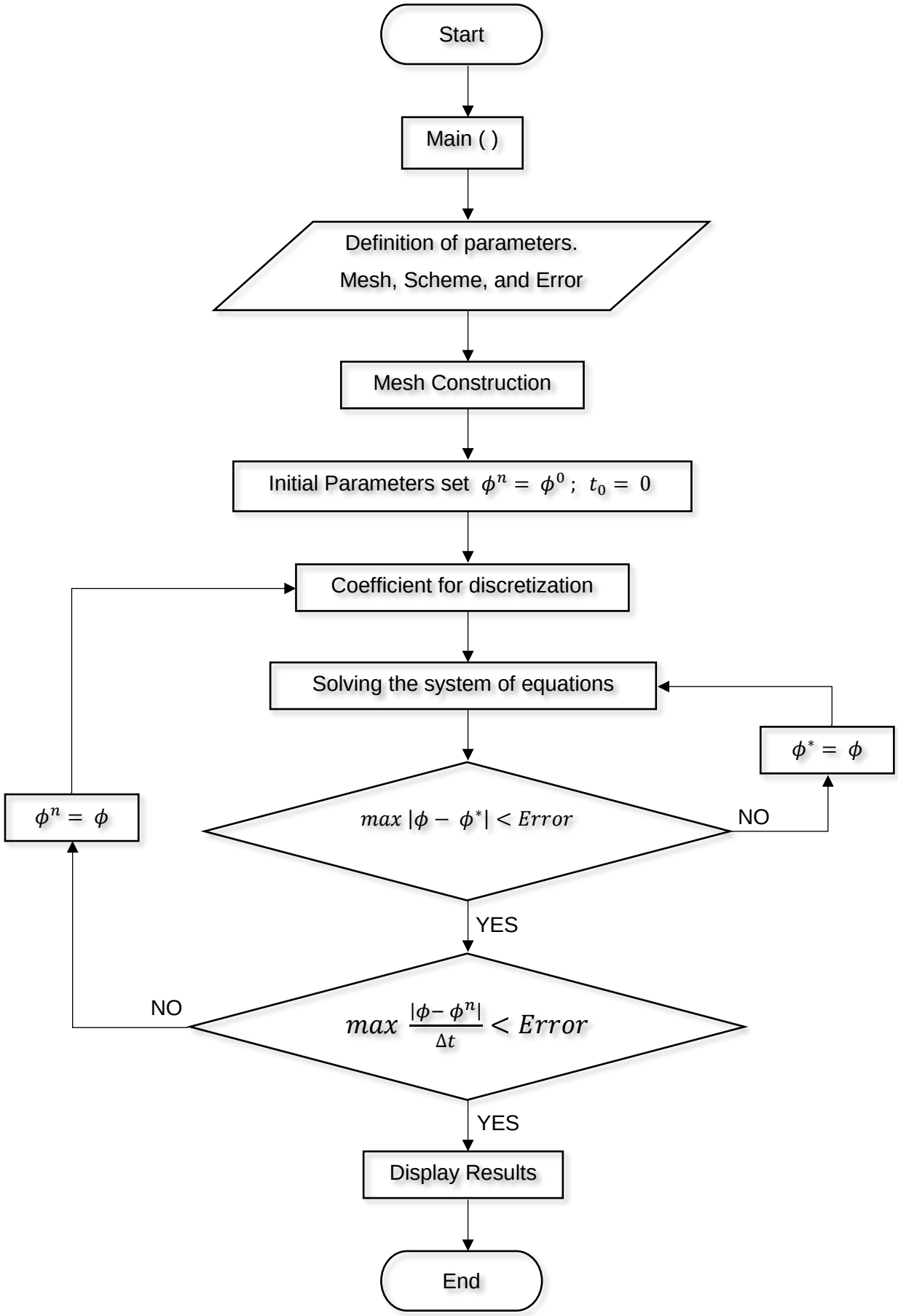


Figure 9. Diagonal Flow: Algorithm flow chart

2.4.1.5. Results

The main goal now it is to analyze the consequences, or the effects that the mesh parameters, high and low order schemes, and the solver method developed have on the results to understand the physical meaning of the problem and to determine the numerical effects that may appear and affect the physical solution. Therefore, the results will be presented in three categories. Mesh Size, Schemes variation, and solver method. In each category, the parameter will be evaluated, thus vary, and the rest will be kept constant. Finally, some conclusion will be drawn according to these specifications.

- **Mesh Size**

The analysis of the mesh influence on the results will be done by comparing the results for three mesh sizes. [50x50], [100x100], and [200x200]. The solver method used was line-by-line, scheme used was QUICK, Δt = 0.01, and Error = 1e-7. These last three parameters were constant during the study because its influence will be further study. One of the key points to analyze the effects of mesh size is how this can affect the diffusive and convective effects on the results. These terms can be affected by the mesh parameters and the values of ρ/Γ, that in this case are 10,10^3, and 10^6. For low values of ρ/Γ the convective and diffusive term tend to be similar. Therefore, the diffusive effect is highly seen on the results. On the other hand, high values of ρ/Γ will result in low diffusive effect because the convective term is much greater than the diffusive one.

That is the reason why the results on this section will be presented just for ρ/Γ = 10 and ρ/Γ =.10^6, so it could be analyzed the lowest and the highest convective-diffusive effects. *Figure 10* shows the expected results for this case for high Peclet numbers, and the scheme of how was evaluated the value of ϕ along the inverse diagonal of the flow.

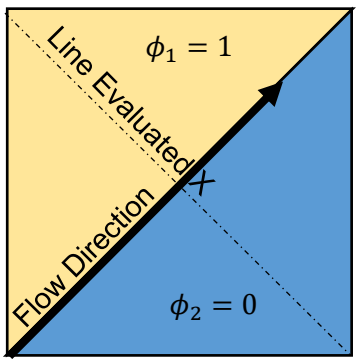


Figure 10. Diagonal Flow: Expected result for Peclet ≈ ∞

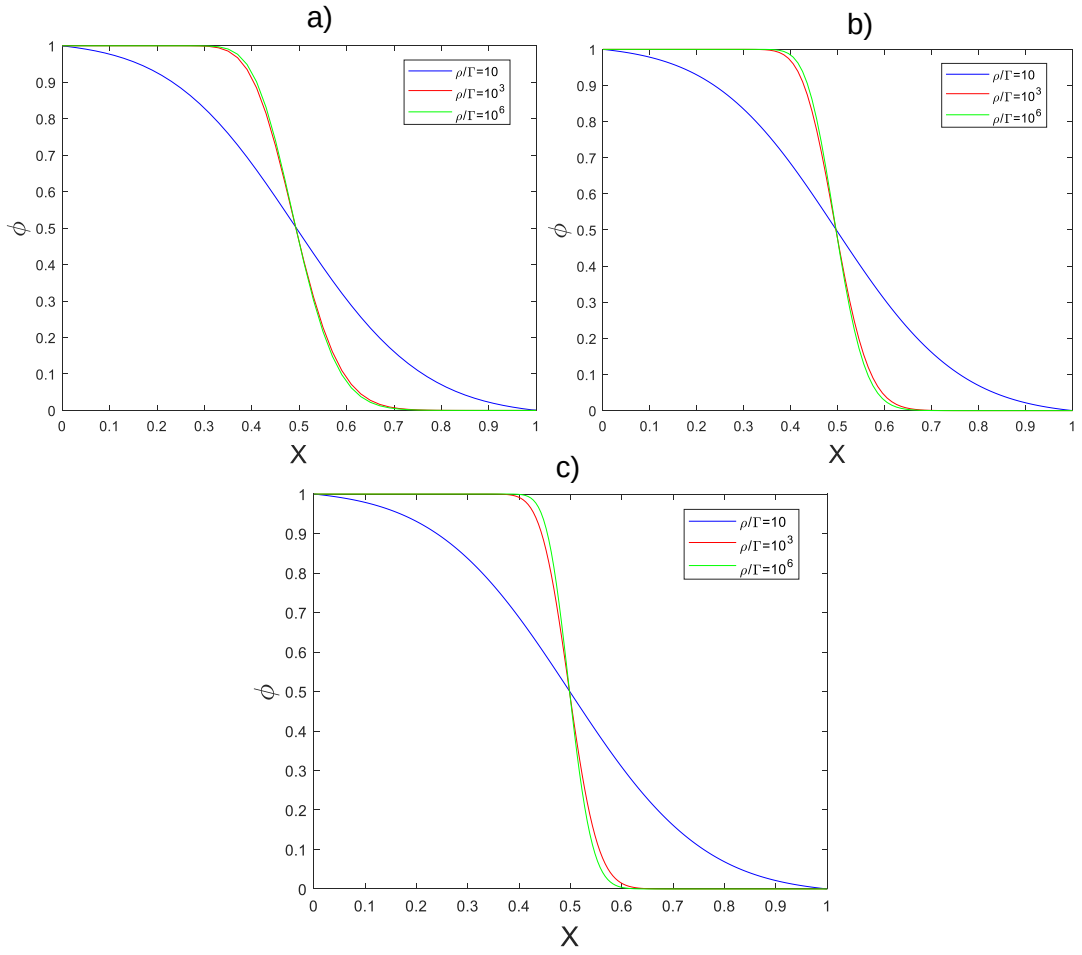


Figure 11. Diagonal Flow: ϕ vs X. a) [50x50], b) [100x100], c) [200x200]

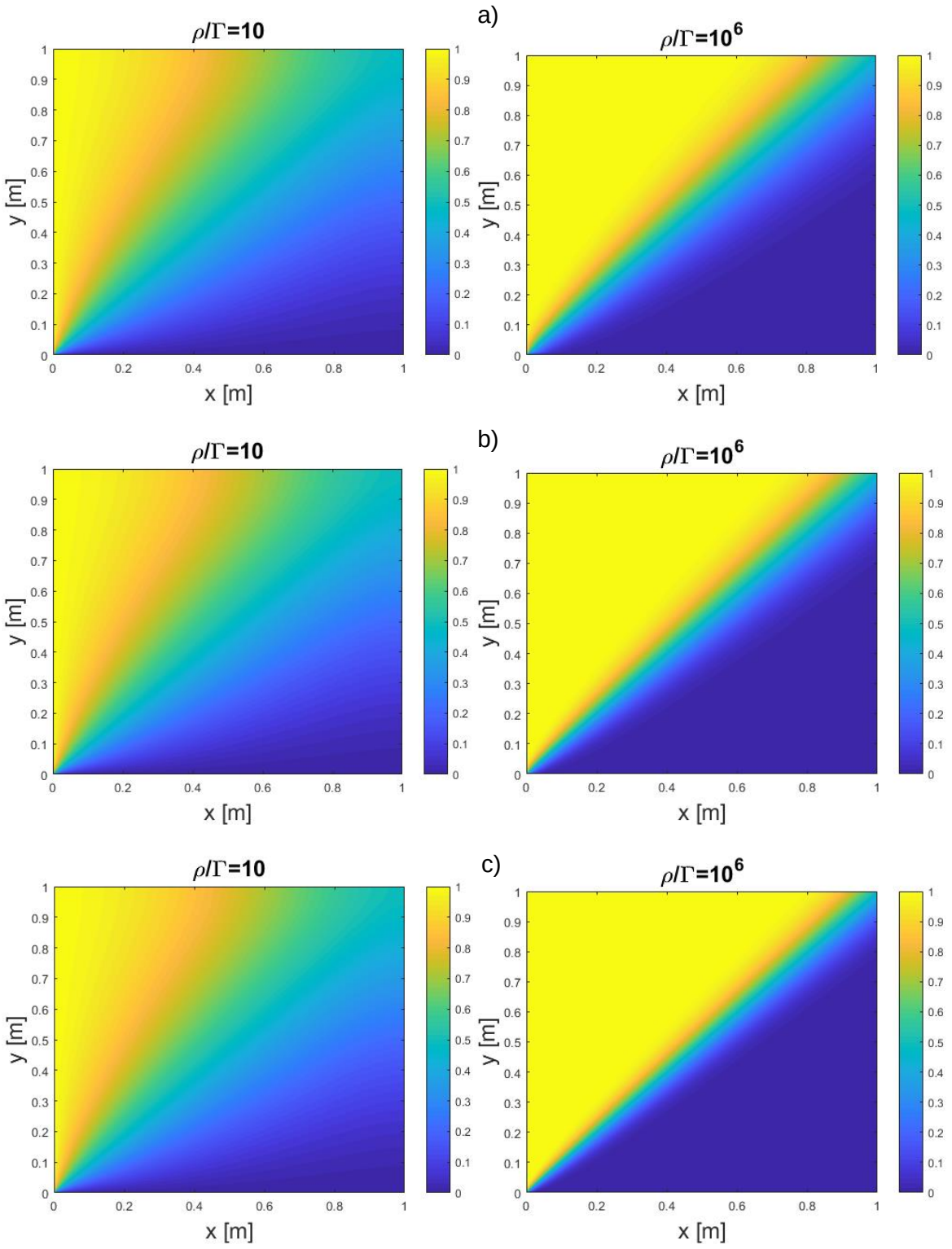


Figure 12. Diagonal Flow: ϕ in the domain. a) [50x50], b) [100x100], c) [200x200]

Figure 11 shows the evolution of ϕ through the evaluated diagonal. For Peclet numbers close to infinite the value is expected to suddenly pass from 1 to 0. It could be seen that the values have a smooth transition around a half of the domain. *Figure 12* shows

graphically the diffusive effect expected in ρ/Γ equals to 10 and 10^6, where the diffusive effect is highly notable in the lowest value ρ/Γ.

Comparing *Figure 12 a, b and c*, it could be seen how the diffusive effect on a finer mesh is less notable, especially for the value of $\rho/\Gamma = 10^6$. Furthermore, *Figure 11* shows the different steeps due to the mesh sizes. Expected results would imply a total vertical steep. However, the diffusive effect creates the variation on the steep.

Finally, the finest mesh gives the least diffusive effect, since the convective term is much greater presented on *Figure 11.a*, the steep on *Figure 11.c* is again grater and closer to be vertical.

- **Scheme Variation**

Following the same path, the analysis of the scheme influence on the results will be done by comparing the results for three schemes. One low-order scheme Upwind, and two high-order schemes Second-order Upwind, and QUICK. The solver method used was line-by-line, mesh size [200x200], Δt = 0.01, and Error 1e-7. These last three parameters were constant during the study. The results of ρ/Γ = 10 and 10^6 will be analyzed in order to compare for the lowest and highest value. Time and iterations to convergence was analyzed to evaluate the time and computational cost.

Somewhat variance among the time of convergence can be seen so far, however a deeper comparison between the schemes will be done. *Table 10* summarize and organize the results of time for each scheme. Consequently, *Figure 13* shows and compare the results for the schemes (Upwind, Second-order upwind, and QUICK) graphically to further draw conclusion about the comparison.

Time of resolution (Diagonal Flow, [200x200], error = 1e-7, Δt = 0.01)						
Scheme	**Upwind**		**Upwind2**		**QUICK**	
Solver	**ρ/Γ**	**Time[s]**	**ρ/Γ**	**Time[s]**	**ρ/Γ**	**Time[s]**
I-b-I	10	1168	10	1179	10	1276
	10^3	177	10^3	180	10^3	186
	10^6	156	10^6	262	10^6	171
	Total	1501	Total	1621	Total	1633

Table 10. Diagonal Flow: Results of convergence time per each scheme

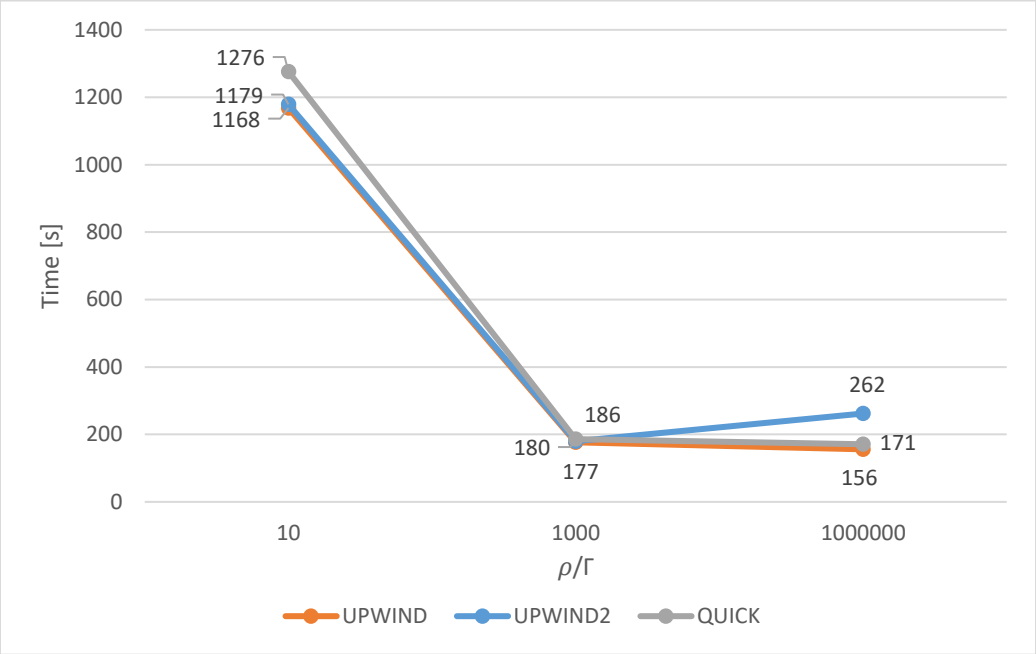


Figure 13. Diagonal Flow: Comparison of convergence time for different schemes

- **Solver Method**

It is time now to analyze the influence of the solver method on the results. It will be studied by comparing the results for two solver methods, Line-by-line and point-by-point. The scheme used was QUICK, mesh size [200x200], Δt = 0.01, and Error 1e-7. These last three parameters were constant during the study. In order to evaluate the effectivity and efficiency of the solver methods the time and iterations to convergence was assessed as well. Furthermore, for the line by line analysis, two different directions were analyzed. As explained in *Chapter 2.3.2* the direction of the "sweeping" process can affect the convergence of the method. Therefore, there is line-by-line method by rows in the positive x direction (left-to-right), and line-by-line by columns in the positive y direction (down-to-up).

In order to summarize the results a set of tables *Table 11, 12, 13* are presented among *Figure 14, 15, 16* that compare the convergence time for each solver method using QUICK scheme. Although the initial process was to use a defined mesh, during the process the mesh size and the schemes had some impact on the results. Thus, it was decided to do two studies more with two other mesh sizes [100x100] and [50x50] and the three schemes.

Time of resolution (Diagonal Flow, [200x200], error = 1e-7, Δt = 0.01)							
Scheme		**Upwind**		**Upwind2**		**QUICK**	
		ρ/Γ	**Time[s]**	**ρ/Γ**	**Time[s]**	**ρ/Γ**	**Time[s]**
Solver	**p-b-p**	10	1202	10	1186	10	1265
		10^3	189	10^3	188	10^3	205
		10^6	170	10^6	274	10^6	187
		Total	1561	Total	1648	Total	1657
	l-b-l (Row+)	10	2257	10	2014	10	2094
		10^3	227	10^3	208	10^3	233
		10^6	200	10^6	273	10^6	196
		Total	2684	Total	2495	Total	2523
	l-b-l (Column+)	10	1168	10	1179	10	1276
		10^3	177	10^3	180	10^3	186
		10^6	156	10^6	262	10^6	171
		Total	1501	Total	1621	Total	1633

Table 11. Diagonal Flow: Time resolution comparison for solver methods [200x200]

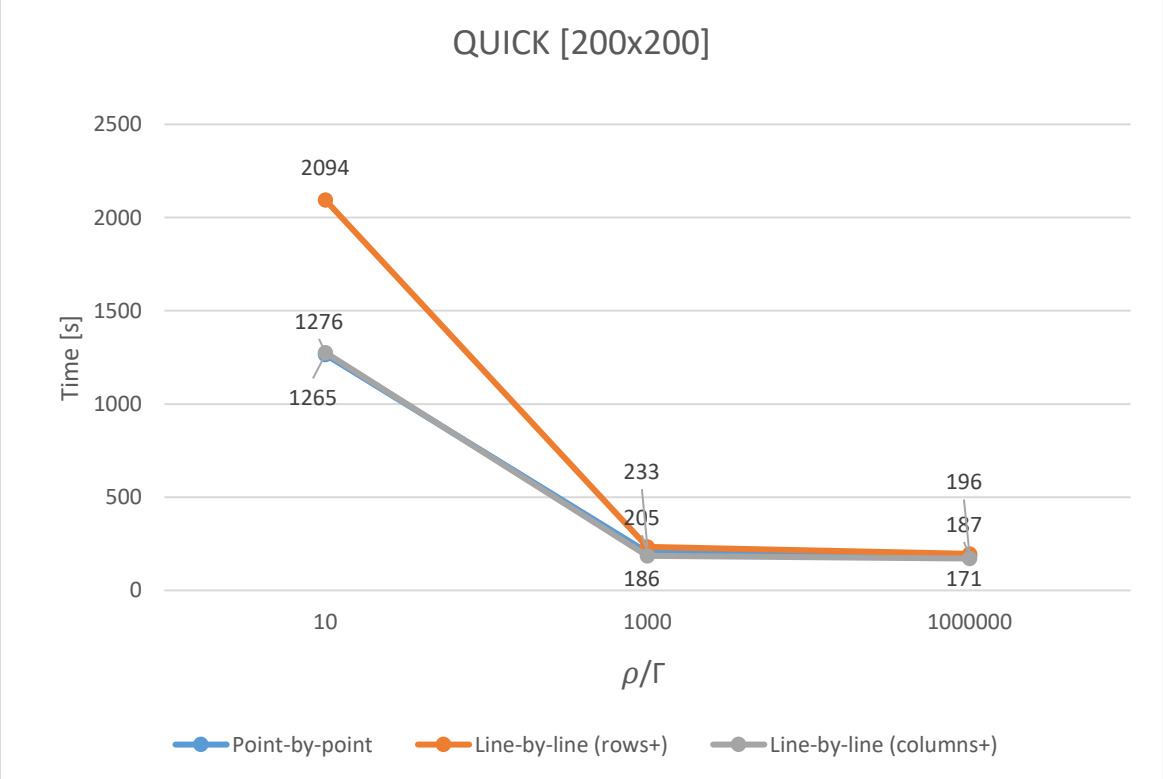


Figure 14. Diagonal Flow: Comparison convergence time for different solver methods [200x200]

Time of resolution (Diagonal Flow, [100x100], error = 1e-7, Δt = 0.01)							
Scheme		**Upwind**		**Upwind2**		**QUICK**	
Solver		**ρ/Γ**	**Time[s]**	**ρ/Γ**	**Time[s]**	**ρ/Γ**	**Time[s]**
	p-b-p	10	154	10	180	10	168
		10^3	44	10^3	43	10^3	44
		10^6	43	10^6	52	10^6	45
		Total	241	Total	275	Total	257
	l-b-l (Row+)	10	200	10	210	10	220
		10^3	43	10^3	47	10^3	46
		10^6	41	10^6	58	10^6	52
		Total	284	Total	315	Total	318
	l-b-l (Column+)	10	156	10	162	10	159
		10^3	41	10^3	41	10^3	44
		10^6	39	10^6	51	10^6	45
		Total	236	Total	254	Total	248

Table 12. Diagonal Flow: Time resolution comparison for solver methods [100x100]

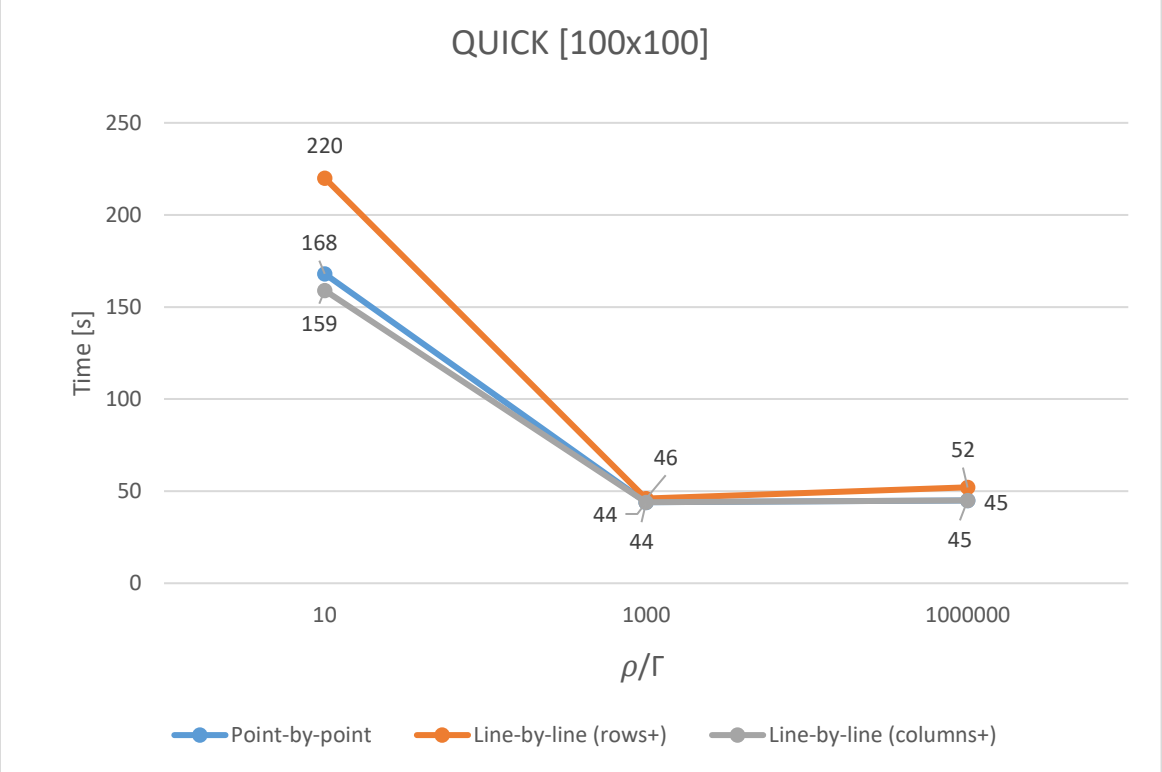


Figure 15. Diagonal Flow: Comparison convergence time for different solver methods [100x100]

Time of resolution (Diagonal Flow, [50x50], error = 1e-7, Δt = 0.01)							
Scheme		**Upwind**		**Upwind2**		**QUICK**	
		ρ/Γ	**Time[s]**	**ρ/Γ**	**Time[s]**	**ρ/Γ**	**Time[s]**
Solver	**p-b-p**	10	27	10	27	10	30
		10^3	11	10^3	10	10^3	11
		10^6	10	10^6	10	10^6	12
		Total	49	Total	47	Total	53
	l-b-l (Row+)	10	30	10	32	10	34
		10^3	10	10^3	11	10^3	11
		10^6	10	10^6	11	10^6	12
		Total	50	Total	54	Total	57
	l-b-l (Column+)	10	26	10	27	10	30
		10^3	10	10^3	10	10^3	11
		10^6	10	10^6	11	10^6	12
		Total	46	Total	48	Total	53

Table 13. Diagonal Flow: Time resolution comparison for solver methods [50x50]

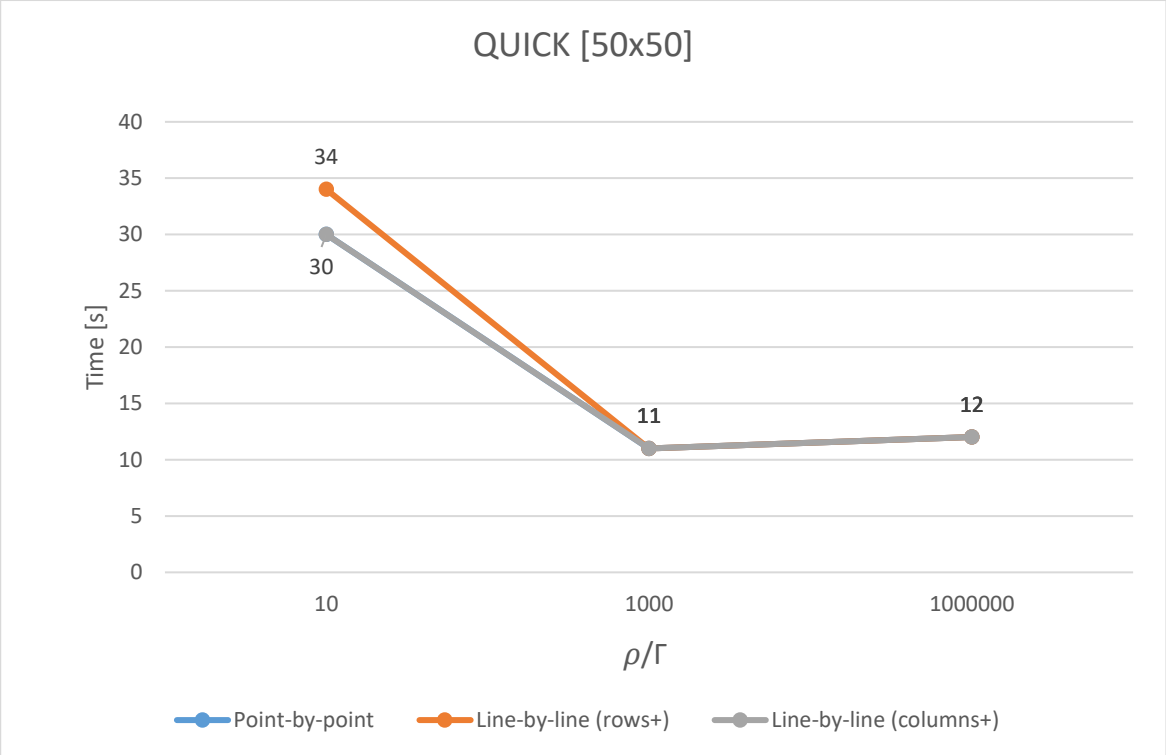


Figure 16. Diagonal Flow: Comparison convergence time for different solver methods [50x50]

2.4.1.6. Discussion of the Results

In conclusion, the influence of three parameters of the discretization of the convection-diffusion (Mesh size, scheme, and solver method) applied to a case of a diagonal flow was analyzed and are presented in the following three main conclusions:

- The size of the mesh does affect the solution by incrementing the diffusive effect for coarse meshes. In *Figure 12* it could be seen how the finer the mesh is, the lesser diffusive effect is shown. It also could be seen how at high values of ρ/Γ the diffusive term become less important than the convective; hence, the diffusive effect is minimized.

- The results showed that the accuracy of the schemes applied to this case is quite similar and accurate enough. However, analyzing other parameters, such as time, iteration and convergence, the high-order schemes seem to last more and use more computational memory. The low-order scheme Upwind was more efficient because the computational work is less than for those high-order as could be seen in *Tables 11, 12, 13*. Also, it could conclude that for large values of ρ/Γ, hence large Peclet number, Upwind scheme is more efficient and accurate since this method calculates the convective terms with the upwind value, this means that it overestimates the diffusive terms. Furthermore, it seems that the value of ρ/Γ influences the time of convergence, for high ρ/Γ the time convergence is much lower than for low ρ/Γ. As a consequence, it could be concluded that the less important the diffusive coefficient is, the faster the solution converge. This is due to the fact that the transportation of the property is faster for the convective term, so the solution converges faster.

- Solver method was mostly influenced by the mesh size, and in the case of the line-by-line by the direction of the "sweeping". It was not influenced much by schemes. Comparing *Figures 14, 15, 16* it could be concluded how the line-by-line in the y positive direction is faster and use less computational resource. This is due mainly because the boundary conditions are transported faster to the middle nodes. For coarse meshes the solver does not highly influence the results.

2.4.2. Smith-Hutton Problem

2.4.2.1. Objective

The second case of study can be applied, this problem is proposed by the CTTC as well, and it consist of the transportation of a variable in a constant semi-circular velocity field in steady state. The main goal is to verify the code generated for rapidly-changing boundary conditions. The control volume is a rectangle and both low-order and high-order schemes will be tested.

2.4.2.2. Problem Definition

This case considers a constant velocity field on 2D domain. The domain is split into two squares forming a rectangle. The flow field *Eq. 75* is entering from the base of the first square, following a semi-circular velocity field and finally exiting by the base of the second square. The scheme is shown in *Figure 17.*

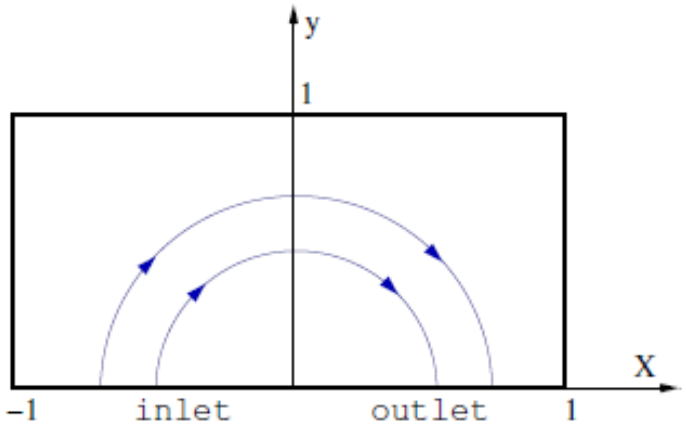


Figure 17. Smith-Hutton: Scheme for the problem and velocity field[7]

The velocity field then is given by *Eq. 75*[7].

$$u(x,y) = 2y \cdot (1 - x^2)$$
$$v(x,y) = -2x \cdot (1 - y^2) \qquad \textit{(Eq. 75)}$$

And boundary conditions given by *Eq. 76*[7]:

$$\phi = 1 + \tanh\big(\alpha(2x+1)\big) \qquad y = 0\,;x\,(-1.0) \quad \text{inlet}$$
$$\frac{\partial \phi}{\partial y} = 0 \qquad y = 0\,;x\,(-1.0) \quad \text{outlet} \qquad \textit{(Eq. 76)}$$
$$\phi = 1 - \tanh(\alpha) \qquad \text{Elsewhere}$$
$$\text{Where } \alpha = 10$$

In order to gain insights on the convective and diffusive effects, three different values of ρ/Γ will be used: $10, 10^3, 10^6$.

2.4.2.3. Discretization

The discretization method is exactly the one used on the prior case *Chapter 2.4.1.3,* so it will not be explained again here. However, the boundary conditions and velocity field are different; therefore, the new table of coefficients for boundary grid nodes are shown in *Table 14.* The rest of the coefficients are presented in *Tables 8 and 9*.

Coefficient	Values of boundary conditions		
	West, East and North Nodes	South Nodes	
		Inlet	Outlet
a_E	0	0	0
a_W	0	0	0
a_N	0	0	1
a_S	0	0	0
a_P	1	1	1
b	$1-\tanh(\alpha)$	$1+\tanh\big(\alpha(2x+1)\big)$	0
b_{dc}	0	0	0

Table 14. Smith-Hutton: Coefficients for boundary conditions

Two important comments should be highlighted about the solution of this case. First of all, even though the problem asks for the steady condition, the terms that contain time dependency were taken into account because this study will analyze the time and computational cost for different numerical schemes and solver methods. This leads to the second consideration, and it is the fact that the case was solved using line-by-line and point-by-point method explained in *Chapter 2.3*, in the same way, the other three schemes different from the previous problem were used: Upwind, Froom method and SMART explained in *Chapter 2.2*. (Find values $\phi_e^{HRS}, \phi_w^{HRS}, \phi_n^{HRS}, \phi_s^{HRS}$ in *Table 2*)

2.4.2.4. Algorithm

The algorithm used for the previous exercise *Diagonal Flow* is the same that will be used here; therefore, the algorithm flow chart could be found in *Figure 9*. This code is implemented three times, each time for a different value of ρ/Γ specified before. See *Attachment I – 2.1* for the algorithm implemented.

2.4.2.5. Results

The same three parameters used for the previous problem will be evaluated for the Smith-Hutton problem. Mesh Size, Schemes variation, and solver method. In each category, the parameter will be evaluated, thus vary, and the rest will be kept constant. The reference solution provided by the problem *Table 15* and *Figure 18* will be used to compare the results and finally draw some conclusions according to these specifications.

X	ρ/Γ		
	10	**10^3**	**10^6**
0.0	1.989	2.0000	2.000
0.1	1.402	1.9990	2.000
0.2	1.146	1.9997	2.000
0.3	0.946	1.9850	1.999
0.4	0.775	1.8410	1.964
0.5	0.621	0.9510	1.000
0.6	0.480	0.1540	0.036
0.7	0349	0.0010	0.001
0.8	0.227	0.0000	0.000
0.9	0.111	0.0000	0.000
1.0	0.000	0.0000	0.000

Table 15. Smith-Hutton: Reference values of ϕ [7]

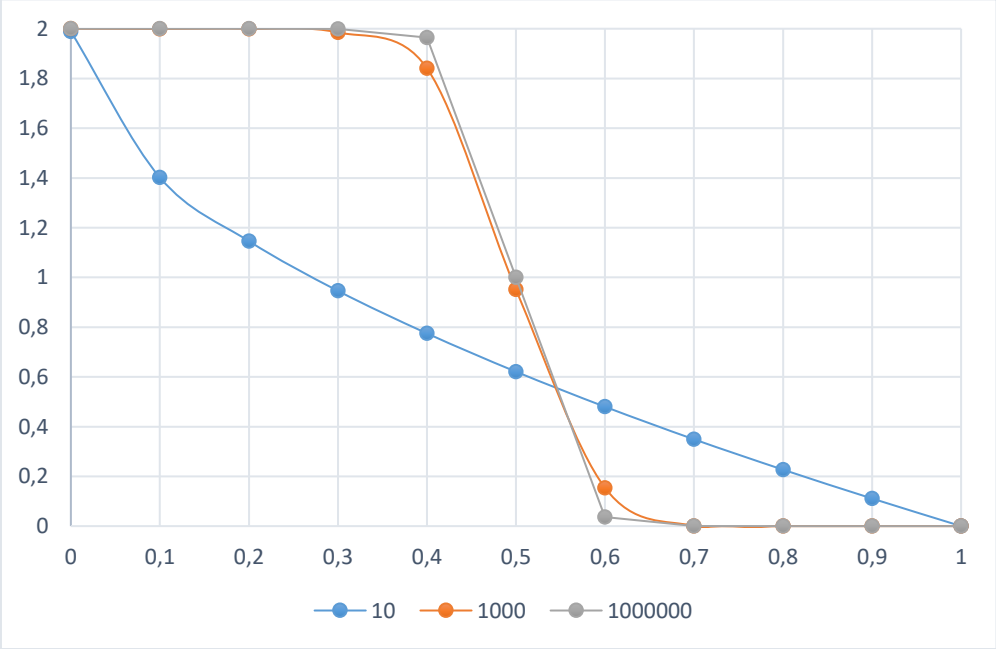


Figure 18. Smith-Hutton: Expected graphic solution

- **Mesh Size**

Three different meshes were used. [100x50], [200x100], and [300x150]. The high-order scheme Fromm method was used as a constant scheme and the solver method used was line-by-line, Δt = 0.01, and Error 1e-7.

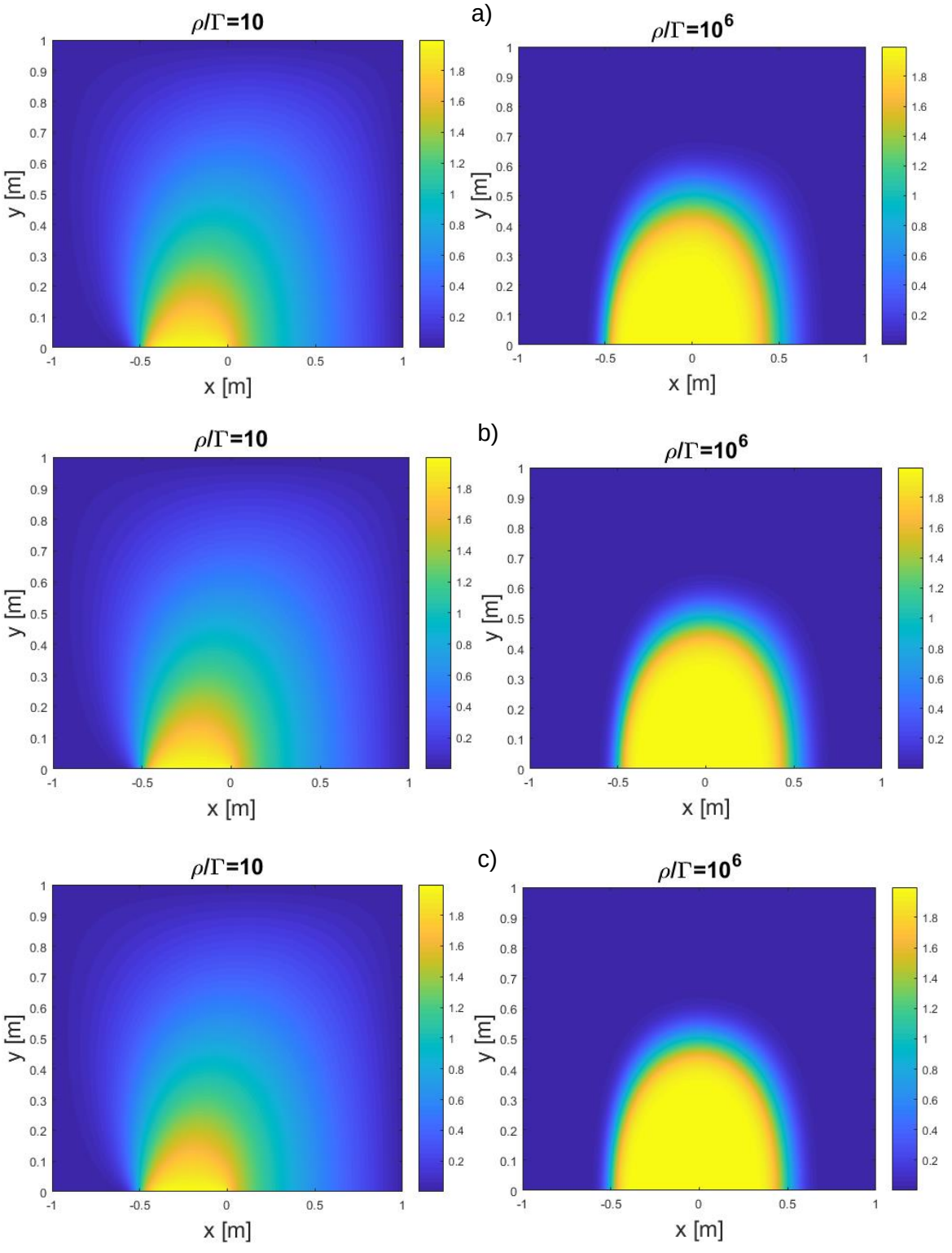


Figure 19. Smith-Hutton: ϕ in the domain a) [100x50], b) [200x100], c) [300x150]

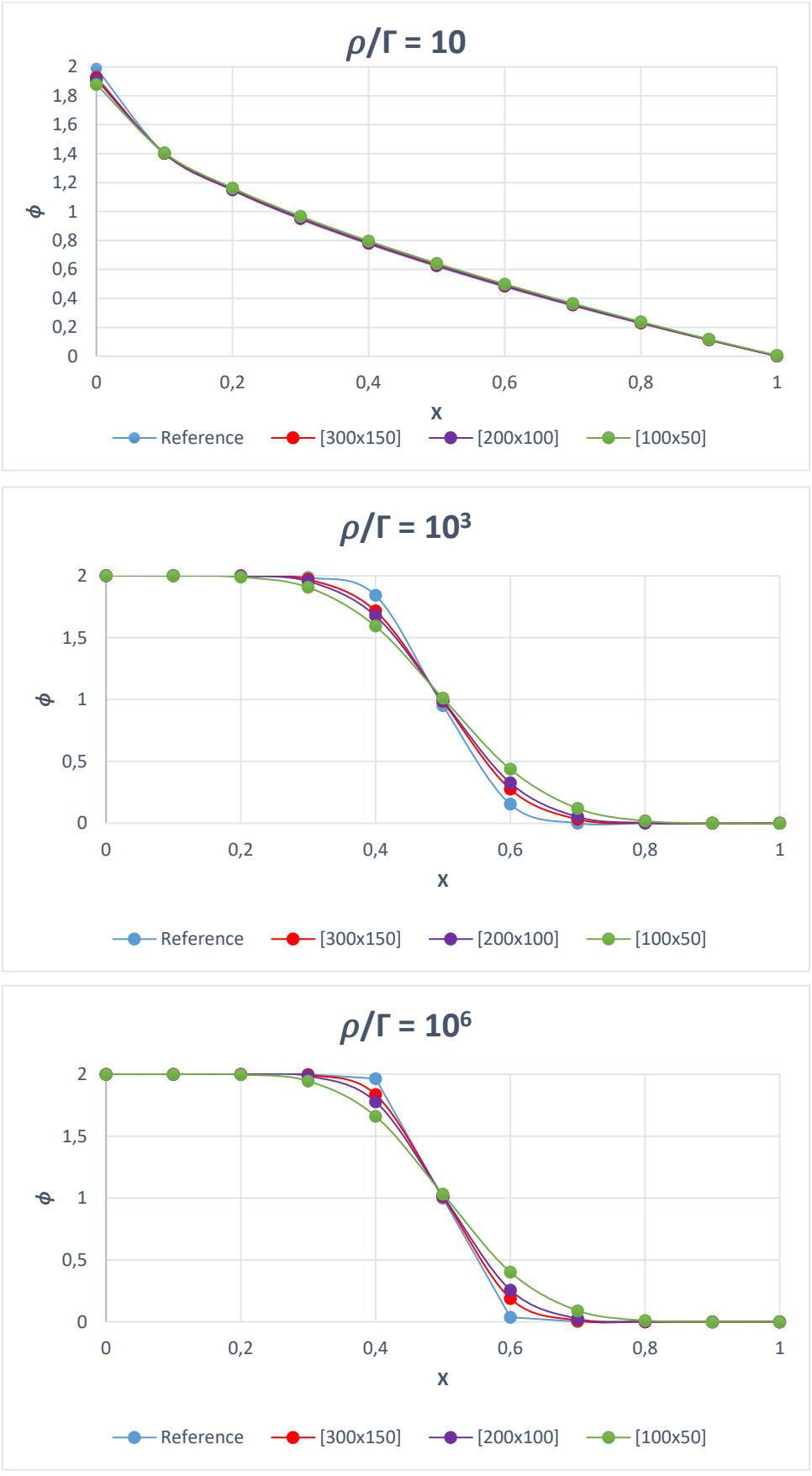


Figure 20. Smith-Hutton: ϕ vs X Comparison of ρ/Γ between mesh sizes

- **Scheme Variation**

Following the same path, the analysis of the scheme influence on the results will be done by comparing the results for three schemes. One low-order scheme Upwind, and two high-order schemes Fromm method, and SMART. The solver method used was line-by-line, mesh size [300x150], Δt = 0.01, and Error 1e-7. These last three parameters were constant during the study. The results of ρ/Γ = 10 and 10^6 will be analyzed in order to compare for the lowest and highest value. Time and iterations to convergence was analyzed to evaluate the time and computational cost.

Time of resolution (Smith-Hutton, [300x150], error = 1e-7, Δt = 0.01)						
Scheme	**Upwind**		**Fromm**		**SMART**	
Solver	**ρ/Γ**	**Time[s]**	**ρ/Γ**	**Time[s]**	**ρ/Γ**	**Time[s]**
l-b-l	10	1139	10	1471	10	1478
	10^3	507	10^3	696	10^3	723
	10^6	599	10^6	874	10^6	7190
	Total	2245	Total	3041	Total	9391

Table 16. Smith-Hutton: Results of convergence time per each scheme

Table 16 and *Figure 21* summarize the results of time convergence for the three schemes. It is important to note the great amount of time taken by the SMART scheme for high values of ρ/Γ. It was almost 900% more time than Fromm method and upwind. On the other hand, the low-order scheme Upwind took around 25% less time than the high-order schemes for each value of ρ/Γ.

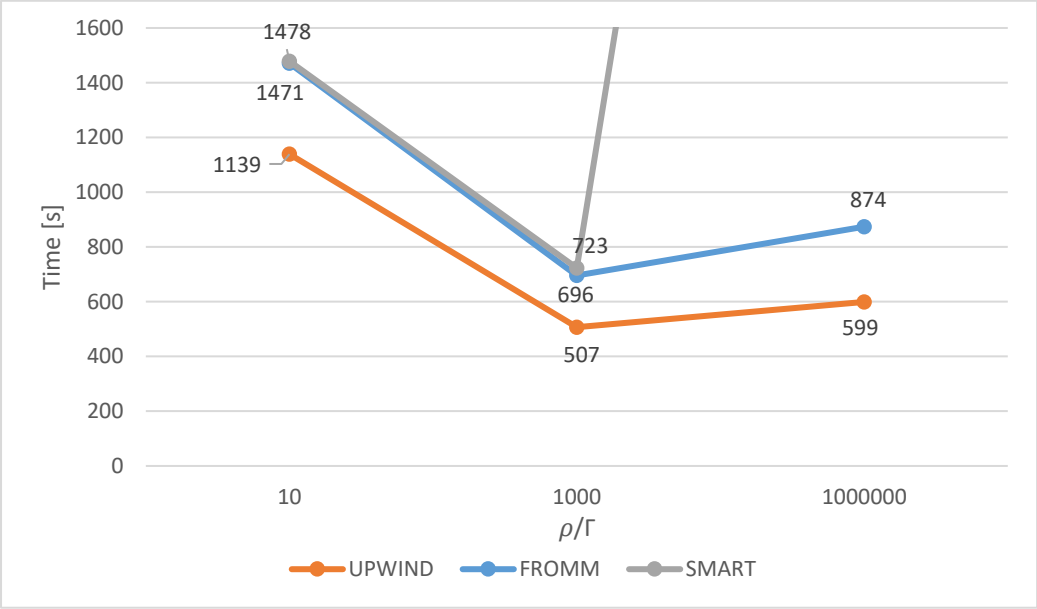


Figure 21. Smith-Hutton: Comparison of convergence time for different schemes

- **Solver method**

In order to summarize the results a set of tables *Table 17, 18, 19* are presented among *Figure 46, 47, 48* that compare the convergence time for each solver method using Fromm method. Although the initial process was to use a defined mesh, during the process the mesh size and the schemes had some impact on the results. Thus, it was decided to do two studies more with two other mesh sizes [200x100] and [150x50] and the three schemes.

Time of resolution (Smith-Hutton, [300x150], error = 1e-7, Δt = 0.01)							
Scheme		**Upwind**		**Fromm**		**SMART**	
Solver	**p-b-p**	**ρ/Γ**	**Time[s]**	**ρ/Γ**	**Time[s]**	**ρ/Γ**	**Time[s]**
		10	1141	10	1287	10	1659
		10^3	474	10^3	628	10^3	854
		10^6	552	10^6	772	10^6	7267
		Total	2167	Total	2687	Total	9780
	l-b-l (Row+)	10	1962	10	2052	10	2231
		10^3	525	10^3	679	10^3	747
		10^6	584	10^6	802	10^6	7422
		Total	3071	Total	3533	Total	10400
	l-b-l (Column+)	10	1139	10	1471	10	1478
		10^3	507	10^3	696	10^3	723
		10^6	599	10^6	874	10^6	7190
		Total	2245	Total	3041	Total	9391

Table 17. Smith-Hutton: Time resolution comparison for solver methods [300x150]

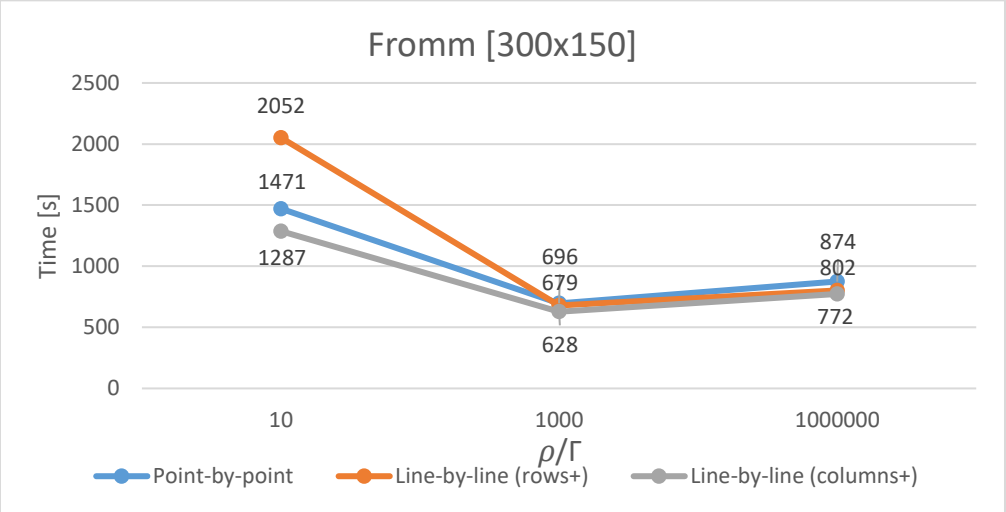


Figure 22. Smith-Hutton: Comparison convergence time for different solver methods [300x150]

Time of resolution (Smith-Hutton, [200x100], error = 1e-7, Δt = 0.01)							
Scheme		**Upwind**		**Fromm**		**SMART**	
		ρ/Γ	**Time[s]**	**ρ/Γ**	**Time[s]**	**ρ/Γ**	**Time[s]**
Solver	**p-b-p**	10	402	10	428	10	433
		10^3	219	10^3	261	10^3	282
		10^6	241	10^6	328	10^6	3892
		Total	862	Total	1017	Total	4607
	l-b-l (Row+)	10	648	10	574	10	622
		10^3	249	10^3	281	10^3	298
		10^6	258	10^6	431	10^6	3857
		Total	1157	Total	1196	Total	4777
	l-b-l (Column+)	10	396	10	418	10	472
		10^3	241	10^3	259	10^3	275
		10^6	274	10^6	319	10^6	3795
		Total	911	Total	996	Total	4542

Table 18. Smith-Hutton: Time resolution comparison for solver methods [200x100]

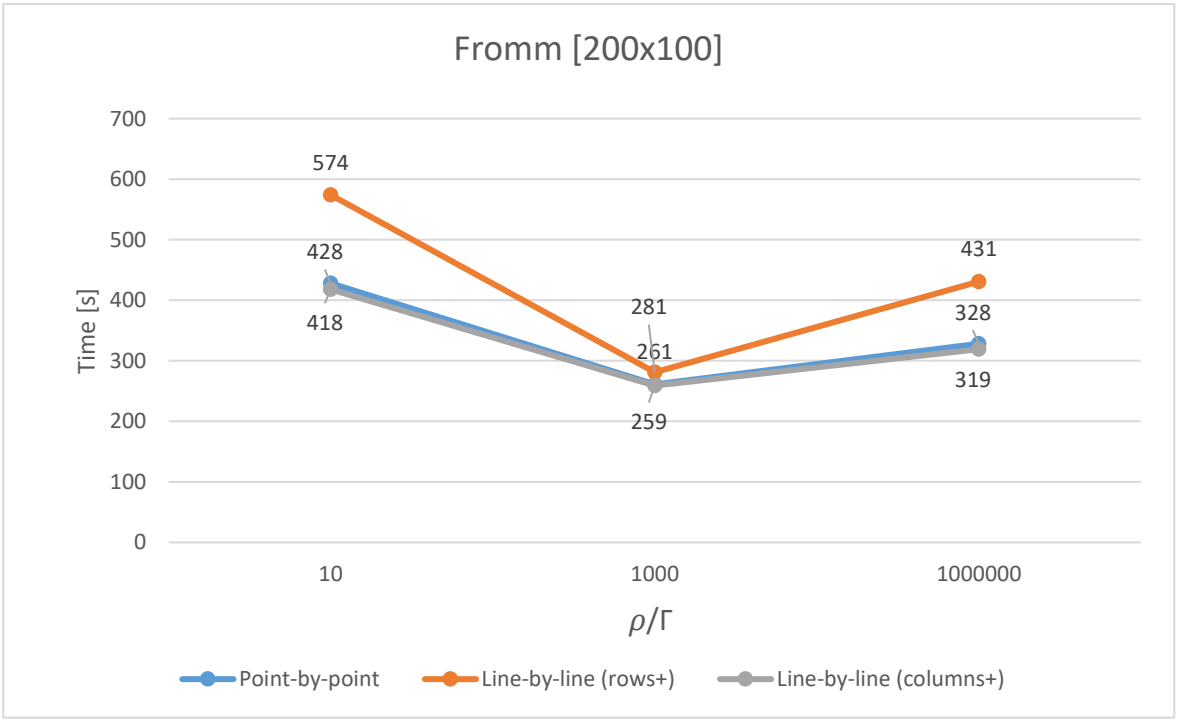


Figure 23. Smith-Hutton: Comparison convergence time for different solver methods [200x100]

Time of resolution (Smith-Hutton, [100x50], error = 1e-7, Δt = 0.01)							
Scheme		**Upwind**		**Fromm**		**SMART**	
		ρ/Γ	**Time[s]**	**ρ/Γ**	**Time[s]**	**ρ/Γ**	**Time[s]**
Solver	**p-b-p**	10	192	10	221	10	259
		10^3	59	10^3	64	10^3	69
		10^6	65	10^6	75	10^6	NC-8.7e-6*
		Total	316	Total	360	Total	
	I-b-I (Row+)	10	145	10	170	10	180
		10^3	53	10^3	60	10^3	63
		10^6	66	10^6	74	10^6	1745
		Total	264	Total	304	Total	1988
	I-b-I (Column+)	10	79	10	75	10	77
		10^3	56	10^3	63	10^3	66
		10^6	67	10^6	79	10^6	1886
		Total	202	Total	217	Total	2029

Table 19. Smith-Hutton: Time resolution comparison for solver methods [100x50]

*NC = No convergence. The solver method using SMART scheme did not converge, but the error (8.7e-6) was close to the error required (1e-7).

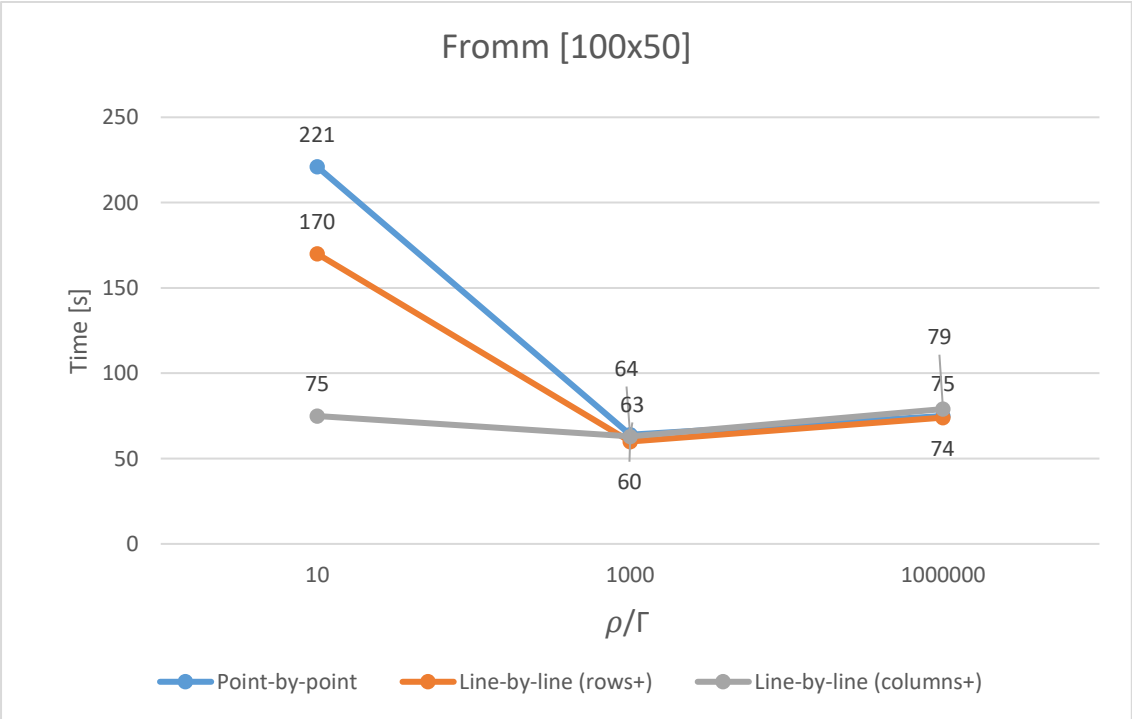


Figure 24. Smith-Hutton: Comparison convergence time for different solver methods [100x50]

2.4.2.6. Discussion of the Results

In conclusion, the influence of three parameters of the discretization of the convection-diffusion (Mesh size, scheme, and solver method) applied to the Smith-Hutton problem was analyzed and are presented in the following three main conclusions:

- It could be proved again that the size of the mesh does affect the solution by incrementing the diffusive effect for coarse meshes. In *Figures 33*, *35* and *37* it could be seen how the finer the mesh is, the lesser diffusive effect is shown.

- The results showed that the accuracy of the schemes applied to this case is quite similar and accurate enough. However, analyzing other parameters, such as time, iteration and convergence, the high-order schemes are more unstable that low-order. In *Table 19* it could be seen how SMART scheme did not converge due to the coarse mesh. SMART scheme took about 600 % more time for high values of ρ/Γ than Fromm method and Upwind scheme. It should be highlighted though the fact that it not converged, but the error was approximately 8.6 x10-6. Which is a small error, but never reached the error desired that was 1x10-7. Since that, it could be concluded that these schemes when used in coarse meshes the convergence criteria is a parameter to be highly analyzed. Furthermore, it seems that the value of ρ/Γ influences the time of convergence, for high ρ/Γ the time convergence is much lower than for low ρ/Γ. As a consequence, it could be concluded that the less important the diffusive coefficient is, the faster the solution converge. This is due to the fact that the transportation of the property is faster for the convective term, so the solution converges faster.

- Solver method was mostly influenced by the mesh size, and in the case of the line-by-line by the direction of the "sweeping". It was not influenced much by schemes.

2.4.3. Two-dimensional Unsteady Heat Transfer Conduction Problem

2.4.3.1. Objective

The application of the convection-diffusion equation to real problems is relevant in the study and analysis of the numerical parameters. Therefore, a transient-conduction heat transfer problem is proposed by the CTTC. Since it is a pure conduction heat transfer problem in a solid body, it is going to be a pure diffusion problem.

2.4.3.2. Problem Definition

This problem consists of a very large rod made of four materials each one with different physical properties. It is going to be analyze in a transient condition due to its time-dependent boundary conditions up until 1000 seconds has passed[8]. The problem requires to compute the temperature at specific points of the domain during the time (Pa [0.65, 0.56] and Pb [0.74, 0.72]), assuming initial temperature in the entire body equal to 8 °C. *Figure 25* shows the scheme of the problem with its material distribution.

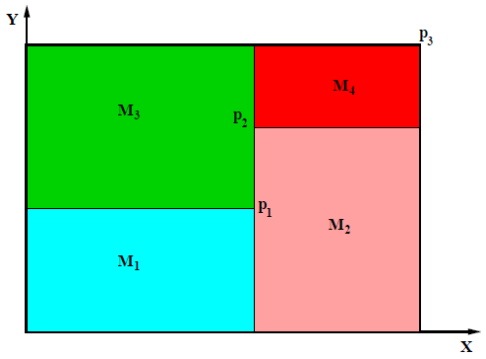


Figure 25. 2D Heat Transfer: Scheme of the rod and its material distribution*[8]*

Point	X [m]	Y[m]
P1	0.5	0.4
P2	0.5	0.7
P3	1.10	0.8

Table 20. 2D Heat Transfer: Points for the dimensions of the rod*[8]*

Material	ρ [kg/m^3]	Cp [J/Kg K]	k [W/m K]
M1	1500	750	170
M2	1600	770	140
M3	1900	810	200
M4	2500	930	140

Table 21. 2D Heat Transfer: Material physical properties[8]

Cavity Wall	Boundary Condition
Bottom	Isotherm at $T = 23\,°C$
Top	Uniform $Qflow = 60$ W/m Length
Left	Contact with fluid at $Tg = 33\,°C$, heat transfer coefficient 9 [W/m K]
Right	Uniform temperature $T = 8 + 0.005t\,°C$ (t in seconds)

Table 22. 2D Heat Transfer: Boundary conditions for the heat conduction transfer problem[8]

Since the rod is considered a long body in length, it could be considered that the heat transfer along its axial axis is null. Hence, it is considered a two-dimensional heat transfer problem.

2.4.3.3. Discretization

The conduction heat transfer in a solid body could be analyzed with the convection-diffusion equation *Eq. 4*; however, considering that the convective effect is not present because there is not transportation of mass within the body. Therefore, it is considered a pure-diffusion problem. The discretization method is then similar to the one done in *Chapter 2.1* but not regarding the convective term. Besides, the problem now has physical meaning, and that means, that now ϕ is the TEMPERATURE, which is the property it is being analyzed. In priory studies, it was assumed an implicit temporal model, but for this study it will be analyzed other models to compare the effects and differences of this models. Then, when the time-dependent term is discretized, a new coefficient appears β, and define the weighting factor or the importance of the property (Temperature in this case) of the prior time step to find the new value of that property. In other words, a value of $\beta = 1$ means that the new value of the property depends exclusively of the value in the at that same time step. This model is known as implicit model. If $\beta = 0$ says that the new value depends exclusively of the old or storage values of the property and it is called explicit model. A 50-50 % of importance with the new and the prior value of the property ($\beta = 0.5$) is called Crack-Nicolson model[4].

Discretizing the new equation and applying the temperature variable it is obtained:

$$\frac{\rho VCp}{\Delta t}\left(T_p - T_p^n\right) = \beta \sum \dot{Q}_p + (1-\beta)\sum \dot{Q}_p^n \qquad \textit{(Eq. 77)}$$

Where $\dot{Q}_p$ is the heat fluxes on the control volume defined by the central node P.

The mesh will be generated exactly as the ones done on the studies before. *Figure 1, 3 and 8*. This way it is possible to have the same hypothesis to discretize the equations. There are two important aspects to evaluate in this problem. First is the fact that the mesh

should be arrange correctly, so the control volume face fits with the joints of the materials. In that way, the fluxes can be correctly found, and the problem will be easier to solve. The second consideration is the act that the fluxes on one of the faces between two materials will be exposed to two different physical properties. Therefore, the total flux needs to be calculated considering both properties. The first problem is easily solved by imposing condition to the mesh construction. The control volume dimension will be attached to this condition and the program will ensure that one of the faces is at the same position of the material intersections. The second problem will just apply for the thermal conductivity λ, because it is the only property which will involve on the heat flux equations. It is solved using the harmonic mean value[4]. *Figure 26* shows the case when a face is between two different material A and B. The fluxes on the face are calculated with equation *Eq. 78* which is described by the Fourier's law.

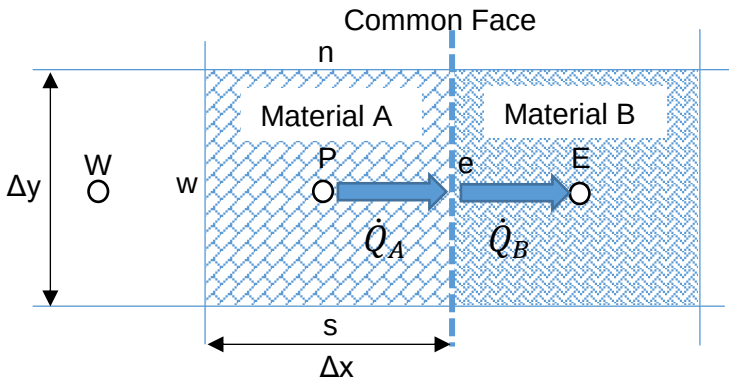


Figure 26. 2D Heat Transfer: Common face shared by two materials

$$\dot{q} = -\lambda \frac{\partial T}{\partial x} \left[{}^{W}/_{m} \right] \qquad \textit{(Eq. 78)}$$

The harmonic mean calculates the total flux on the shared face. Therefore, it involves the property for both materials A and B. The total flux on the face must be equal to zero. Then by applying Fourier's law:

$$\dot{q_A} = -\lambda_A \frac{T_e - T_P}{d_{Pe}} \left[{}^{W}/_{m} \right] \quad ; \quad \dot{q_B} = -\lambda_B \frac{T_E - T_e}{d_{Ee}} \left[{}^{W}/_{m} \right] \qquad \textit{(Eq. 79)}$$

If the fluxes are equal, then the harmonic mean could be found:

$$\dot{q} = \dot{q_A} = \dot{q_B} = \frac{d_{PE}}{\frac{d_{Pe}}{\lambda_A} + \frac{d_{Ee}}{\lambda_B}} \frac{T_E - T_P}{d_{PE}} \left[{}^{W}/_{m} \right]$$

$$\lambda_{A-B} = \frac{d_{PE}}{\frac{d_{Pe}}{\lambda_A} + \frac{d_{Ee}}{\lambda_B}} \qquad \textit{(Eq. 80)}$$

It is possible now to discretize the equation and obtain coefficients for the resolution. Taking Eq. and discretizing it leads to:

$$\sum \dot{Q}_p = \left(-\lambda_E \frac{(T_P - T_E)}{(\delta x)_{PE}} \Delta y - \lambda_W \frac{(T_P - T_W)}{(\delta x)_{PW}} \Delta y - \lambda_N \frac{(T_P - T_N)}{(\delta y)_{PN}} \Delta x] - \lambda_S \frac{(T_P - T_S)}{(\delta y)_{PS}} \Delta x\right) \quad \textit{(Eq. 81)}$$

$$\sum \dot{Q}_p^n = \left(-\lambda_E \frac{(T_P^n - T_E^n)}{(\delta x)_{PE}} \Delta y - \lambda_W \frac{(T_P^n - T_W^n)}{(\delta x)_{PW}} \Delta y - \lambda_N \frac{(T_P^n - T_N^n)}{(\delta y)_{PN}} \Delta x] - \lambda_S \frac{(T_P^n - T_S^n)}{(\delta y)_{PS}} \Delta x\right) \quad \textit{(Eq. 82)}$$

By introducing *Eq. 81* and *Eq. 82* in *Eq. 77*, the middle grid point's coefficient can be found. It is summarized in *Table 23*.

Coefficient	Values for Middle Nodes	
a_E	$\beta\lambda_E \frac{\Delta y}{(\delta x)_{PE}}$	*(Eq. 83)*
a_W	$\beta\lambda_W \frac{\Delta y}{(\delta x)_{PW}}$	*(Eq. 84)*
a_N	$\beta\lambda_N \frac{\Delta x}{(\delta y)_{PN}}$	*(Eq. 85)*
a_S	$\beta\lambda_S \frac{\Delta x}{(\delta y)_{PS}}$	*(Eq. 86)*
a_P	$a_E + a_W + a_N + a_S + \frac{\rho(\Delta x \Delta y)Cp}{\Delta t}$	*(Eq. 87)*
b	$\left[\frac{\rho(\Delta x \Delta y)Cp}{\Delta t} + (1-\beta)\left(-\lambda_E \frac{\Delta y}{(\delta x)_{PE}} - \lambda_W \frac{\Delta y}{(\delta x)_{PW}} - \lambda_N \frac{\Delta x}{(\delta y)_{PN}} - \lambda_S \frac{\Delta x}{(\delta y)_{PS}}\right)\right] T_P^n + (1-\beta)\left(\lambda_E \frac{\Delta y}{(\delta x)_{PE}}\right) T_E^n + (1-\beta)\left(\lambda_W \frac{\Delta y}{(\delta x)_{PW}}\right) T_W^n + (1-\beta)\left(\lambda_N \frac{\Delta x}{(\delta y)_{PN}}\right) T_N^n + (1-\beta)\left(\lambda_S \frac{\Delta x}{(\delta y)_{PS}}\right) T_S^n$	*(Eq. 88)*

Table 23. 2D Heat Transfer: Coefficients for middle grid nodes

For the boundary conditions, *Eq. 89* and *90* are applied. First, it is assumed that the boundary grid nodes do not have volume associated. For the boundary conditions the discretization equation is found then:

Top nodes:

$$0 = \beta\left[-\lambda_S \frac{(T_P - T_S)}{\frac{(\delta y)_{PS}}{2}}\Delta x + \Delta x \cdot Qflow\right] + (1-\beta)\left[-\lambda_S \frac{(T_P^n - T_S^n)}{\frac{(\delta y)_{PS}}{2}}\Delta x + \Delta x \cdot Qflow\right] \quad \textit{(Eq. 89)}$$

Left nodes:

$$0 = \beta\left[-\lambda_E \frac{(T_P - T_E)}{\frac{(\delta x)_{PE}}{2}}\Delta y - \alpha(T_P - T_g)\Delta y\right] + (1-\beta)\left[-\lambda_E \frac{(T_P^n - T_E^n)}{\frac{(\delta x)_{PE}}{2}}\Delta y - \alpha(T_P^n - T_g)\Delta y\right] \quad \textit{(Eq. 90)}$$

The right and bottom grid nodes the temperature is already known, so it is not necessary the finite volume analysis. *Table 24* summarizes the values of the coefficients for the boundary grid nodes.

Coeff	Values of boundary conditions			
	West Nodes	**East Nodes**	**South Nodes**	**North Nodes**
a_E	$\frac{\beta\lambda_E\Delta y}{\frac{(\delta x)_{PE}}{2}} + \alpha\beta\Delta y$	0	0	0
a_W	0	0	0	0
a_N	0	0	0	0
a_S	0	0	0	$\frac{\beta\lambda_S\Delta x}{\frac{(\delta y)_{PS}}{2}}$
a_P	a_E	1	1	a_S
b	$\alpha\Delta y T_g + (1-\beta)\left(-\alpha\Delta y - \frac{\lambda_E\Delta y}{\frac{(\delta x)_{PE}}{2}}\right)T_P^n + (1-\beta)\left(\frac{\lambda_E\Delta y}{\frac{(\delta x)_{PE}}{2}}\right)T_E^n$	$8 + 0.005t$	T_{base}	$\Delta x \cdot Qflow + (1-\beta)\left(-\frac{\lambda_s\Delta x}{\frac{(\delta y)_{PS}}{2}}\right)T_P^n + (1-\beta)\left(\frac{\lambda_S\Delta x}{\frac{(\delta y)_{PS}}{2}}\right)T_S^n$

Table 24. 2D Heat Transfer: Coefficients for boundary conditions

2.4.3.4. Algorithm

With the purpose of graphically explain the code done for this case, *Figure 27* shows the algorithm flow chart that briefly explain the main processes inside the code. The code runs a main code and three sub-codes. the mesh construction, the computation of discretization coefficients and finally the solver. See *Attachment I – 2.1* for the algorithm implemented.

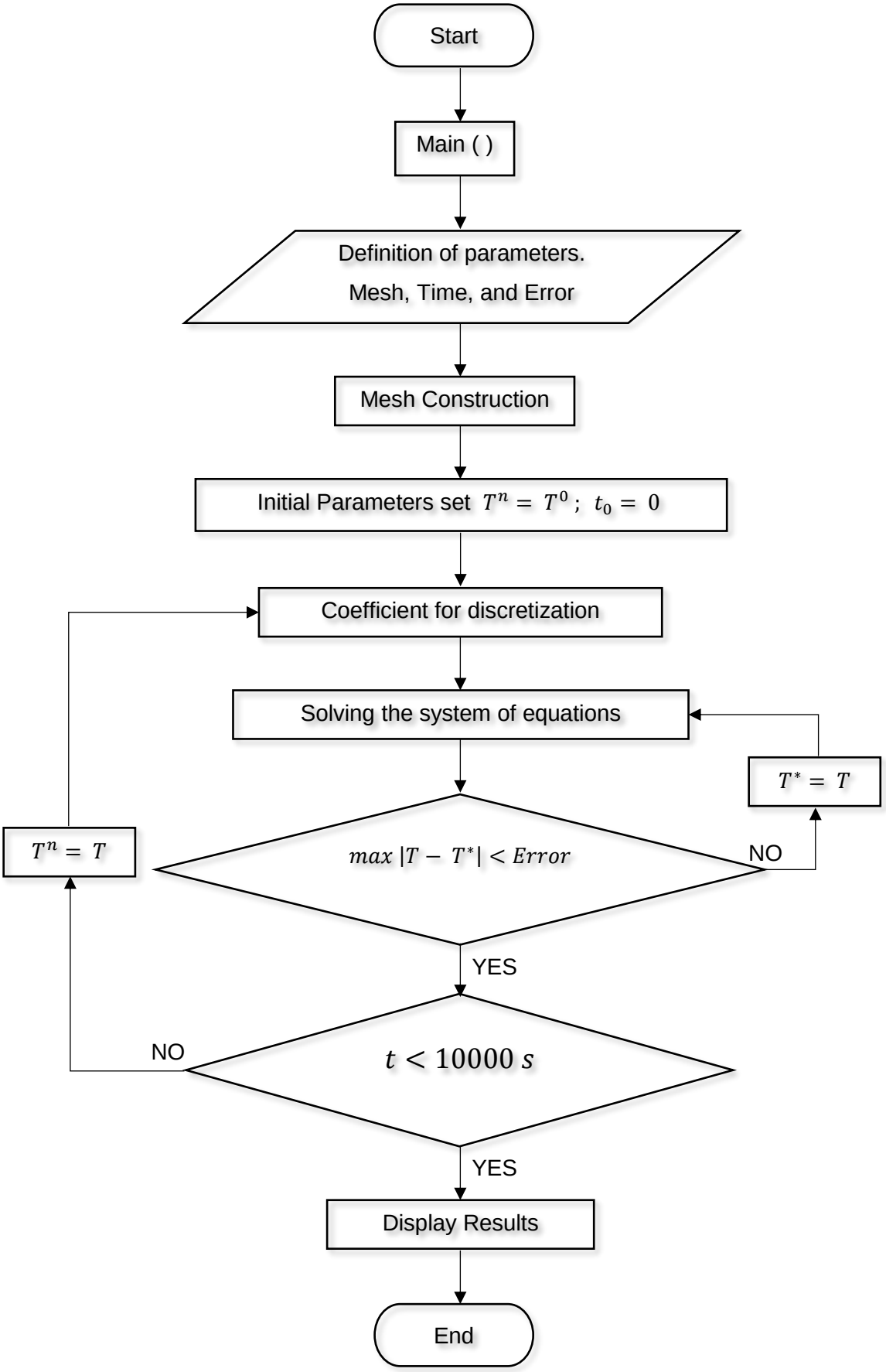


Figure 27. 2D Heat Transfer: Algorithm flow chart

2.4.3.5. Results

The results shown in *Figure 28, 29* and *Table 25* are for the Crank-Nicolson model β = 0.5; however, an analysis with a fully implicit model was realized and the results will be shown in *Figure 30, 31* and *Table 26,* so then it could be compared with the Crank-Nicolson model.

- Crank-Nicolson model, β = 0.5

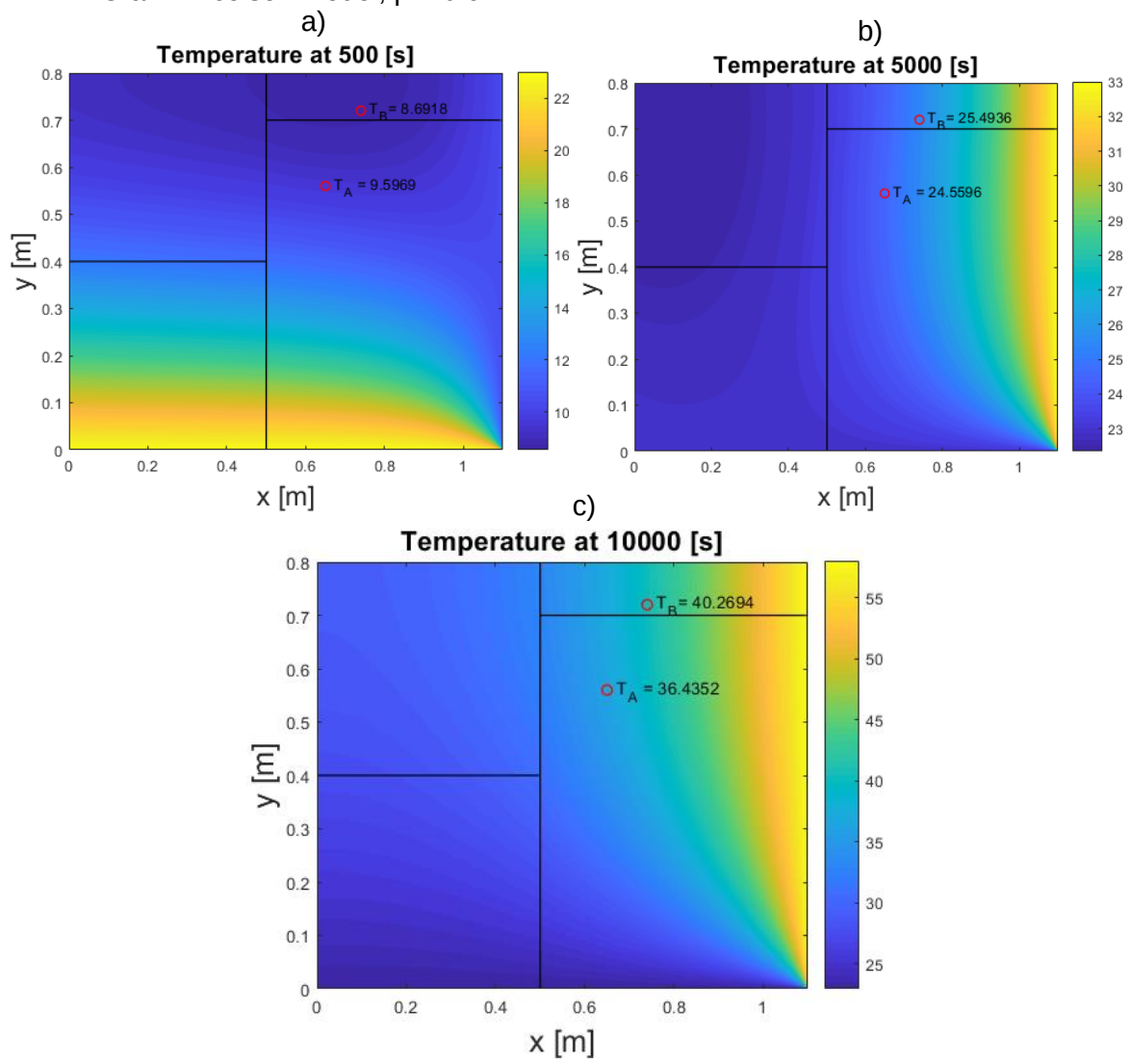


Figure 28. 2D Heat Transfer: Temperature field a) 500 s, b) 5000 s, c) 1000 s, β = 0.5, [220x160]

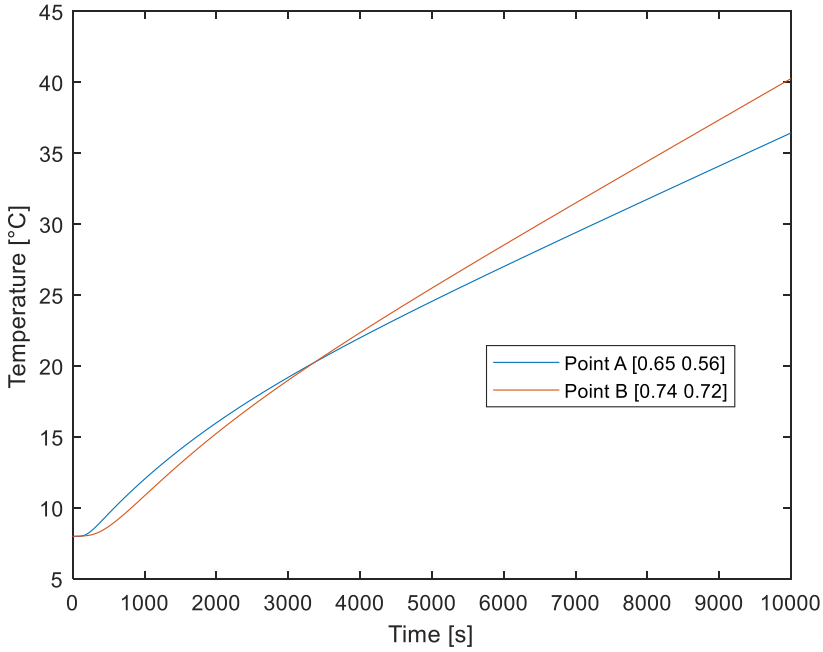


Figure 29. 2D Heat Transfer: Evolution of point A and B, β = 0.5, [220x160]

Temperature evolution for point and B β = 0.5, [200x160]								
Time[s]	**A [°C]**	**B [°C]**	**Time[s]**	**A [°C]**	**B [°C]**	**Time[s]**	**A [°C]**	**B [°C]**
0	8	8	**3200**	19.79	19.7	**6800**	28.93	30.9
10	8	8	**3400**	20.36	20.38	**7000**	29.4	31.49
100	8.01	8.01	**3600**	20.91	21.05	**7200**	29.88	32.08
200	8.17	8.05	**3800**	21.46	21.7	**7400**	30.35	32.67
300	8.57	8.16	**4000**	21.99	22.35	**7600**	30.82	33.26
500	9.6	8.69	**4200**	22.52	22.99	**7800**	31.29	33.85
700	10.62	9.49	**4400**	23.04	23.63	**8000**	31.76	34.43
1000	12.03	10.9	**4600**	23.55	24.26	**8200**	32.23	35.02
1200	12.91	11.8	**4800**	24.06	24.88	**8400**	32.7	35.6
1400	13.74	12.7	**5000**	24.56	25.49	**8600**	33.17	36.19
1600	14.53	13.6	**5200**	25.06	26.11	**8800**	33.63	36.77
1800	15.28	14.4	**5400**	25.55	26.71	**9000**	34.1	37.35
2000	16	15.3	**5600**	26.04	27.32	**9200**	34.57	37.94
2200	16.69	16.1	**5800**	26.53	27.92	**9400**	35.04	38.52
2400	17.35	16.8	**6000**	27.01	28.52	**9600**	35.5	39.1
2600	17.99	17.6	**6200**	27.49	29.12	**9800**	35.97	39.69
2800	18.61	18.3	**6400**	27.97	29.72	**10000**	36.44	40.27
3000	19.2	19	**6600**	28.45	30.31			

Table 25. 2D Heat Transfer: Results of temperature evolution β = 0.5, [220x160]

- Fully implicit model, β = 1

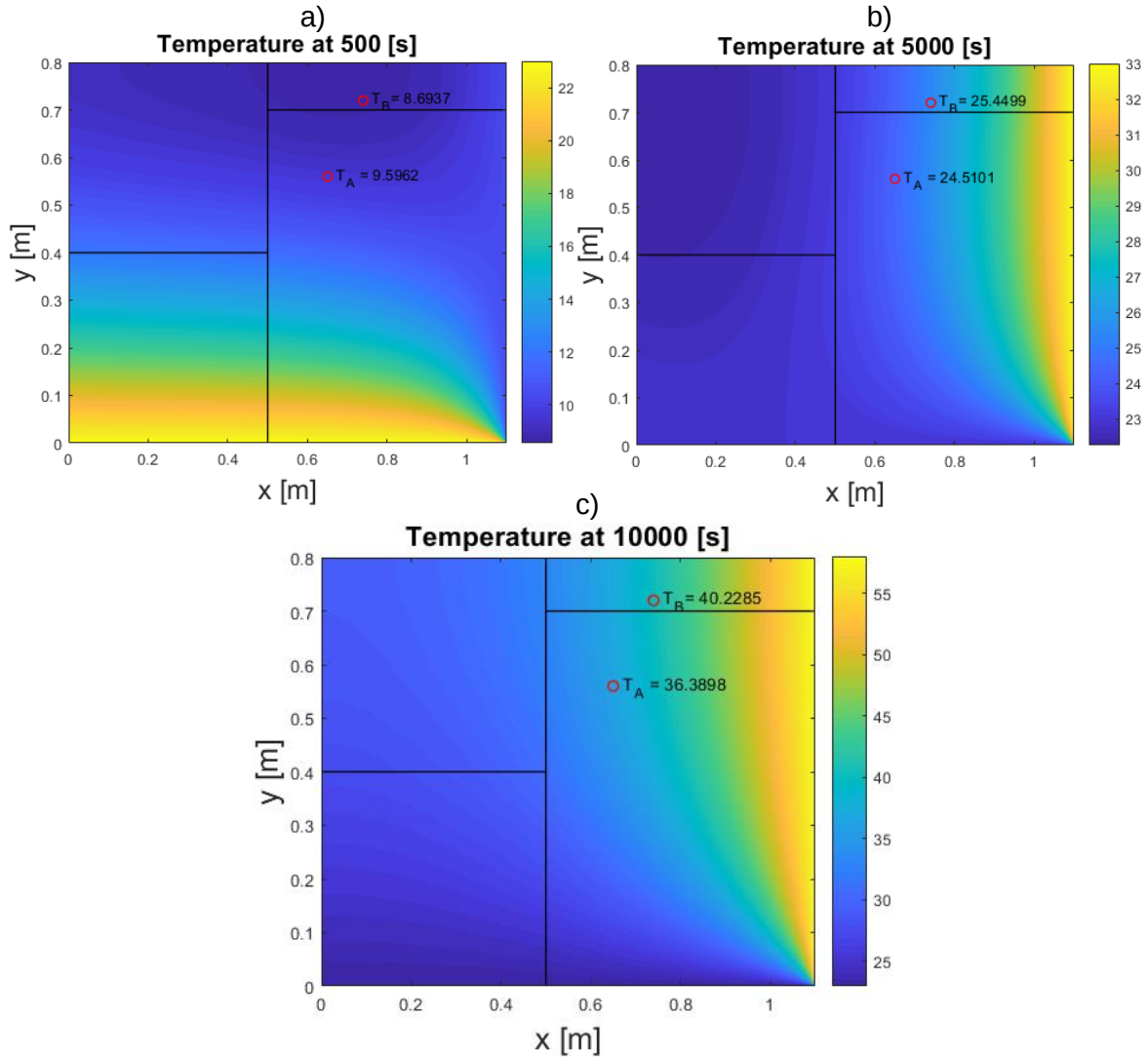


Figure 30. 2D Heat Transfer: Temperature at 500, 5000 and 1000 s, β = 1, [220x160]

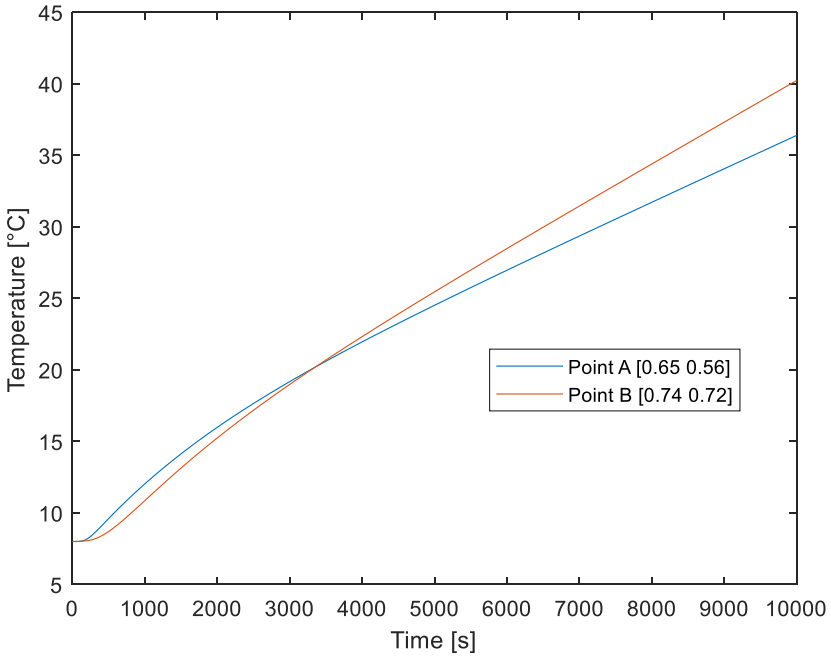


Figure 31. 2D Heat Transfer: Evolution of point A and B, β = 1, [220x160]

Temperature evolution for point and B β = 1, [200x160]								
Time[s]	**A [°C]**	**B [°C]**	**Time[s]**	**A [°C]**	**B [°C]**	**Time[s]**	**A [°C]**	**B [°C]**
0	8	8	**3200**	19.75	19.67	**6800**	28.88	30.86
10	8	8	**3400**	20.32	20.35	**7000**	29.35	31.45
100	8.01	8.01	**3600**	20.87	21.01	**7200**	29.83	32.04
200	8.17	8.05	**3800**	21.42	21.67	**7400**	30.3	32.63
300	8.57	8.16	**4000**	21.95	22.32	**7600**	30.77	33.21
500	9.6	8.69	**4200**	22.48	22.96	**7800**	31.24	33.8
700	10.61	9.49	**4400**	22.99	23.59	**8000**	31.71	34.39
1000	12.02	10.85	**4600**	23.5	24.21	**8200**	32.18	34.97
1200	12.9	11.78	**4800**	24.01	24.83	**8400**	32.65	35.56
1400	13.72	12.69	**5000**	24.51	25.45	**8600**	33.12	36.14
1600	14.51	13.56	**5200**	25.01	26.06	**8800**	33.59	36.73
1800	15.26	14.41	**5400**	25.5	26.67	**9000**	34.06	37.31
2000	15.98	15.23	**5600**	25.99	27.27	**9200**	34.52	37.9
2200	16.66	16.03	**5800**	26.48	27.88	**9400**	34.99	38.48
2400	17.32	16.79	**6000**	26.96	28.48	**9600**	35.46	39.06
2600	17.96	17.54	**6200**	27.44	29.07	**9800**	35.92	39.65
2800	18.57	18.27	**6400**	27.92	29.67	**10000**	**36.39**	**40.23**
3000	19.17	18.97	**6600**	28.4	30.26			

Table 26. 2D Heat Transfer: Results of temperature evolution β = 1, [220x160]

2.4.3.6. Discussion of the Results

First of all, the results are highly accurate regarding the boundary conditions and the conduction heat transfer effects on the domain. The results shows in *Figure 54* and *52* are similar to the results given in the reference [8]. Furthermore, it could be seen the transient effect and how the boundary conditions change the internal heat transfer on the domain. Based on this, it could be concluded that the developed code performs good enough to give accurate results.

- Regarding the fully implicit and Crank-Nicolson model used for this problem, it could be concluded that both gives accurate results. As could be seen in *Tables 25* and *26* the results are almost the same. Difference do not go over $1x10^{-2}$ which is accurate enough for this case. The computational cost for the implicit scheme is much lower than for the Crank-Nicolson model. This is due mainly because in the fully implicit model the storage of prior values is not necessary. On the other hand, for Crack-Nicolson the new values depend of the old or storage values. This means that the values need to be storage for the next time step. Considering that the values and accuracy are similar, it could be concluded that is more convenient the implicit model rather than the computational-expensive Crank-Nicolson.

- Analyzing the results and the temperature patterns among the domain, a conclusion could be drawn about the influence of the boundary conditions and the materials properties. First the boundary conditions have a major effect on the heat transfer within the domain. This could be seen on the temperature evolution in Points A and B. Point A which location is closer to the bottom is hotter earliest time because the temperature effect on the bottom is more important at the beginning. Once the time past, approximately after 3400 seconds, Point B becomes hotter because the temperature at the east wall is greater and point B is closer to it. Furthermore, the material physical properties appear to not influence much on the gradient temperature among the material joints. The patterns do no change suddenly on the joints. This could be due to the values of conductivity of the materials which are not so different.

2.5. Chapter Summary

In this chapter, three main computational parameters involved in the numerical resolution of the convection-diffusion equation have been defined and furtherly analyzed by solving three applied cases of study: two of convection-diffusion effects and one of pure diffusion.

First, the spatial and time discretization of the differential equation was analyzed under some assumptions necessary for the discretization development. The discretization was done using a Finite Volume Method and a structured mesh. Secondly, the numerical schemes of low and high order were defined and studied. Results showed that low order schemes have a lower performance compared to high-order, in other words, less accurate; however, low order schemes have a computational cost much lower than high resolution schemes. Also, high order schemes are more unstable depending on the case, specifically high values of ρ/Γ. Thus, the convergence criteria and time-step utilized must be thoroughly analyzed when using high order schemes.

Consequently, two solver methods were defined and tested in this study: Point-by-point and Line-by-line. Both methods were proven to be effective in most of the problems. However, Line-by-line method was proven to be somewhat more efficient and consume less computational cost than Point-by-point. The computational cost for the cases of study may not appear to be a considerable amount, however, for more difficult and complicated cases it may be a crucial parameter to be aware in numerical resolution.

Lastly, physical conditions such as values of ρ/Γ and mesh sizes were analyzed as well. Results showed that high values of ρ/Γ reduces the diffusive effects, therefore, finer meshes are necessary to achieve convergence. Also, that high convective effects will required of high order numerical schemes in order to obtained accurate results. Furthermore, an interesting pure-diffusion case was analyzed using fully implicit and Crank-Nicolson methods which could proof and validate the algorithms developed and that were applied to the cases of study.

Chapter 3

NAVIER-STOKES EQUATIONS

The Navier-Stokes equations consists of a set of time-dependent equations: the continuity equation of mass conservation, spatial conservation of momentum, and energy conservation[9]. For instance, the Navier-Stokes time-dependent governing equation for an incompressible Newtonian fluid is the following:

$$\rho\left(\frac{\partial \vec{\upsilon}}{\partial t} + \vec{\upsilon}\cdot\nabla\vec{\upsilon}\right) = -\nabla p + \mu\nabla^2\vec{\upsilon} + \vec{f} \qquad \textit{(Eq. 93)}$$

Where u is the fluid's velocity field, ρ is the fluid density, p the pressure, and μ the dynamic viscosity. Also, the term at the left side of the equation correspond to the inertial forces, the first one on the right represents the pressure forces, the next one the viscous forces, and the last term the external forces applied to the fluid[10]. This equation *Eq. 91* will need to be solved along the continuity mass equation *Eq. 92*.

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\vec{\upsilon}) = 0 \qquad \textit{(Eq. 94)}$$

In order to understand the physical meaning of the Navier-Stokes equations, two physical effects are defined using the different terms that composed the equations and were studied priory in *Chapter 2*. First, the CONVECTION TERMS. Convection is the physical process inside a fluid where a property is transported by the systematic motion of the flow. Secondly, the DIFFUSIVE TERMS. Diffusion is the physical process where a property is transported in a fluid by the arbitrary motion of the fluid particles[9]. Nonetheless, not just the Navier-Stokes partial differential equations seem to follow the same pattern of CONVECTION-DIFFUSION terms. Some of the most relevant differential equations on physics have the same generalized scheme, where a dependent variable is defined by convective terms, related to mass fluxes, diffusive terms specific to the meaning of the dependent variable, and finally a so-called source term which is related more to the numerical solutions and external parameters.

This chapter will stablish the main process of the Fractional Step Method, the most common and easiest way for numerical resolution of the Navier-Stokes equations along with the

definition of staggered meshes and finally validate the algorithms by solving two applied cases of study.

3.1. Fractional Step Method

One of the most common methods to solve the incompressible Navier-Stokes equations is the Fractional Step Method (FSM). FSM uses simpler code algorithms and gives better performance than other methods such as SIMPLE-like algorithms[11]. It is also known as projection method. *Eq. 93a, 93b* are the incompressible, constant-viscosity form of the Navier-Stokes equations that are solved using FSM.

$$\rho \frac{\partial \vec{\upsilon}}{\partial t} + (\rho \vec{\upsilon} \cdot \nabla)\vec{\upsilon} = \mu(\Delta \vec{\upsilon}) - \nabla p \qquad \textit{(Eq. 93a)}$$

$$\nabla \cdot \vec{\upsilon} = 0 \qquad \textit{(Eq. 93b)}$$

FSM proposes an innovative method which projects the velocity vectors into a divergence-free velocity space. By using the intermediate velocity (convergence-free velocity field) and solving the Poisson equation *Eq. 102* the intermediate velocity becomes and incompressible flow and then the real velocity field is computed.

In order to find the divergence-free velocity vector, it is necessary to apply the Helmholtz-Hodge theorem which states that a vector field ω bounded in the domain Ω, can be decomposed in a pure gradient field and a divergence-free vector that is parallel to $\delta\Omega$. *Eq. 94* expresses mathematically the theorem and *Figure 32* shows graphically the diverge-free and the parallel field implied in the theorem [11].

$$\omega = a + \nabla\varphi$$
$$\nabla \cdot a = 0 \quad ; \quad a \in \Omega \qquad \textit{(Eq. 94)}$$

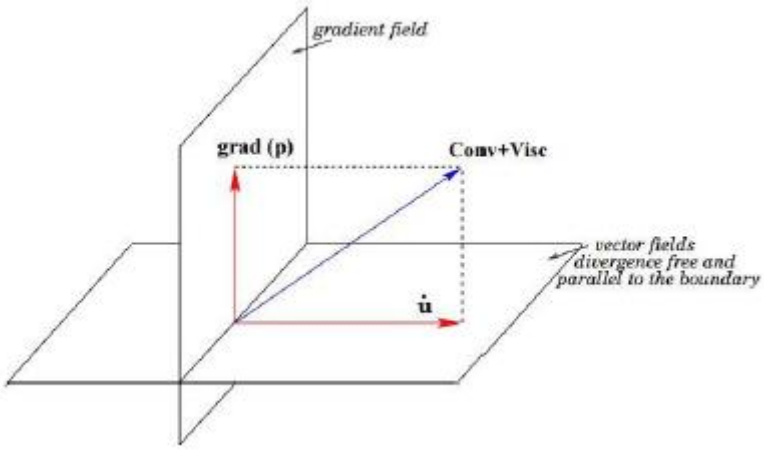


Figure 32. Vector field decomposition Helmholtz-Hodge theorem. Extracted from[11]

3.1.1. Time-integration Method

Once the theorem is defined, it is possible to integrate the incompressible, constant-viscosity equation *Eq. 93a, 93b.*

First, a simpler notation is used in the momentum equations of Navier-Stokes, compiling them and assigning the vector $\boldsymbol{R}(\vec{\upsilon})$.

$$\boldsymbol{R}(\vec{\upsilon}) = \mu(\Delta\vec{\upsilon}) - (\rho\vec{\upsilon}\cdot\nabla)\vec{\upsilon} \qquad \textit{(Eq. 95)}$$

Then, *Eq. 93a* becomes:

$$\rho\frac{\partial\vec{\upsilon}}{\partial t} = \boldsymbol{R}(\vec{\upsilon}) - \nabla p \qquad \textit{(Eq. 96)}$$

It is possible now to time-integrate *Eq. 96*. The first hypothesis will be a fully explicit integration for the continuity equations (left term of *Eq. 96*), therefore:

$$\int_n^{n+1}\rho\frac{\partial\vec{\upsilon}}{\partial t} = \rho\frac{\vec{\upsilon}^{n+1} - \vec{\upsilon}^n}{\Delta t} \qquad \textit{(Eq. 97a)}$$

Consequently, the momentum equations $\boldsymbol{R}(\vec{\upsilon})$ are integrated using a fully implicit second-order Adams-Bashforth scheme. Thus, *Eq. 97b* is obtained:

$$\int^{n+1/2}\boldsymbol{R}(\vec{\upsilon}) = \frac{3}{2}\boldsymbol{R}(\vec{\upsilon})^n - \frac{1}{2}\boldsymbol{R}(\vec{\upsilon})^{n-1} \qquad \textit{(Eq. 97b)}$$

Finally, the semi-discretized incompressible Navier-Stokes equations will be the compilation of *Eq. 97a, 97b*, and the first-order backward Euler scheme for the pressure field. Therefore, *Eq. 98a* is obtained:

$$\rho\frac{\vec{\upsilon}^{n+1} - \vec{\upsilon}^n}{\Delta t} = \frac{3}{2}\boldsymbol{R}(\vec{\upsilon})^n - \frac{1}{2}\boldsymbol{R}(\vec{\upsilon})^{n-1} - \nabla p^{n+1} \qquad \textit{(Eq. 98a)}$$

Time-integrating *Eq. 93b* it is obtained:

$$\nabla\cdot\vec{\upsilon}^{n+1} = 0 \qquad \textit{(Eq. 98b)}$$

Then, following the unique decomposition of the Helmholtz-Hodge theorem *Eq. 94* is introduced to the semi-discretized NS equations *Eq. 98a*:

$$\vec{\upsilon}^p = \vec{\upsilon}^{n+1} + \frac{\Delta t}{\rho}\nabla p^{n+1} \qquad \textit{(Eq. 99)}$$

Where $\vec{\upsilon}^p$ is the intermediate or predictor velocity. It is possible now to transform the original momentum equation to the velocity projection equation:

$$\rho\frac{\vec{\upsilon}^p - \vec{\upsilon}^n}{\Delta t} = \frac{3}{2}\boldsymbol{R}(\vec{\upsilon})^n - \frac{1}{2}\boldsymbol{R}(\vec{\upsilon})^{n-1} \qquad \textit{(Eq. 100)}$$

Once the predictor velocity is defined, a relation for the pressure in function of the intermediate velocity can be derive by applying the divergence operator to *Eq. 99*:

$$\nabla \cdot \vec{\upsilon}^{p} = \nabla \cdot \vec{\upsilon}^{n+1} + \nabla \cdot \left(\frac{\Delta t}{\rho} \nabla p^{n+1}\right) \qquad \textit{(Eq. 101)}$$

Where considering *Eq. 98b*, it is replaced and solved to obtain the Poisson equation for the pressure field.

$$\Delta p^{n+1} = \frac{\rho}{\Delta t} \nabla \cdot \vec{\upsilon}^{p} \qquad \textit{(Eq. 102)}$$

And finally, the real velocity field is given by solving *Eq. 99* for $\vec{\upsilon}^{n+1}$.

$$\vec{\upsilon}^{n+1} = \vec{\upsilon}^{p} - \frac{\Delta t}{\rho} \nabla p^{n+1} \qquad \textit{(Eq. 103)}$$

3.1.2. Staggered Meshes

Staggered meshes were designed to solve a problem when the Navier-Stokes equations are discretized and numerically solved. The checkerboard problem is generated because of the link between the pressure and the velocity fields. In a collocated mesh, the control volumes are represented by the main node, usually named P. if the pressure and the velocity are represented by the same central node P this could converged in velocity field for unphysical pressure distributions. For instance, a 1D case *Figure 33* applying FSM the discretized equation for the velocity field would be:

$$\mathrm{u}^{n+1} = \mathrm{u}^{p} - \frac{\Delta t}{\rho} \left(\frac{{p_E}^{n+1} - {p_W}^{n+1}}{2\delta x}\right) \qquad \textit{(Eq. 104)}$$

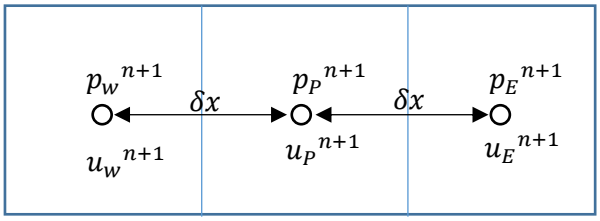


Figure 33. 1D Pressure and velocity discretization leading to the checkerboard problem

From *Eq. 104*, and *Figure 33*, it could be concluded that the discrete approximation of ∇p^{n+1} at the central node P is independent of ${p_P}^{n+1}$ which leads to unphysical pressure field results. If the pressure gradient does not depend on the pressure at node P, that means that the velocity field could converge regardless the pressure at node P and could result in pressure gradients equal to zero. The solution was to use staggered meshes, which solves the checkerboard problem, and they are easy to implement. The staggered mesh will

separate the main nodes used for the pressure field from the horizontal and vertical velocities, so in that way the pressure nodes will always be dependent of the neighbor velocities and pressure nodes. A staggered mesh then contains three different meshes: the main mesh which will be used for the pressure field, one staggered mesh in the horizontal direction for the momentum equation in the X direction (Ux), and one staggered mesh in the vertical direction for the momentum equation in the Y direction (Vy). *Figure 34* shows the basic configuration of a staggered mesh in a structured mesh.

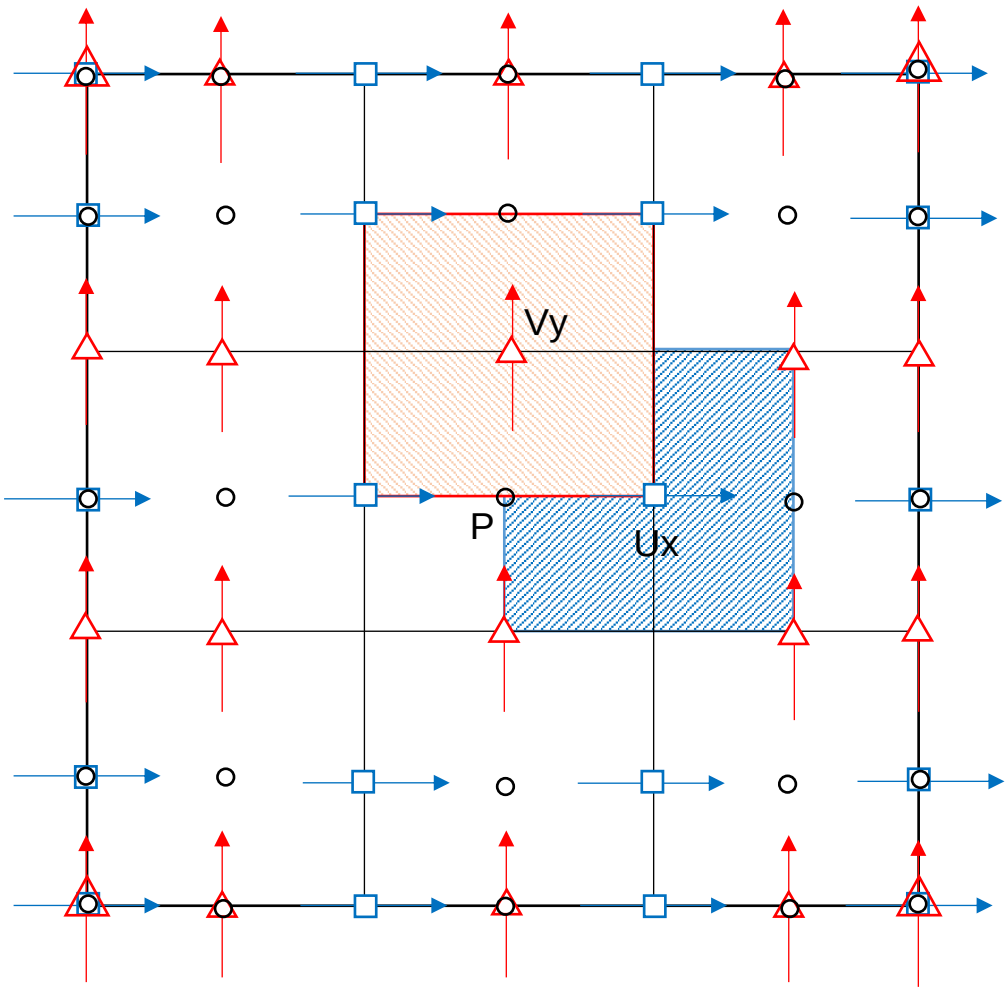


Figure 34. Staggered mesh structure

Where the circles in black represent the central nodes in the main mesh for the pressure control volumes, the blue squares represent the central nodes for the momentum equation Ux in the horizontal staggered mesh, and the red triangles for the momentum equation Vy in the vertical staggered mesh.

3.2. Cases of Study: Fractional Step Method

3.2.1. Driven Cavity

3.2.1.1. Objective

Once FSM is defined, the first case of study can be applied to verify the effectivity of the algorithm. The problem is proposed by CTTC, and it consist of a 2D cavity with one wall moving at constant velocity. Velocity field is requested for different Reynolds numbers.

3.2.1.2. Problem Definition

A bi-dimensional square cavity which its top layer is moving at a constant velocity is the first case of study to validate the FSM algorithm. The other three walls are restricted to no-slip boundary conditions (equal to zero). *Figure 35* shows the general scheme of the problem and *Table 27* the pressure and velocity boundary conditions.

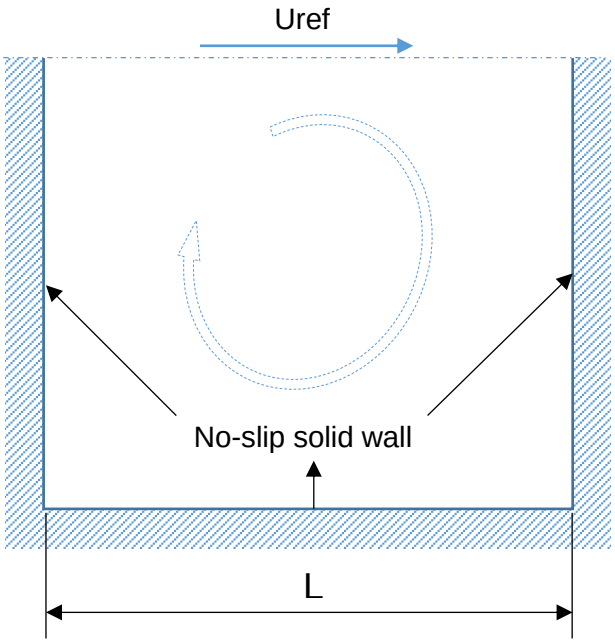


Figure 35. Driven Cavity: General scheme of the case

Cavity	Values of boundary conditions		
	Ux (Horizontal Velocity)	Vy (Vertical Velocity)	Pressure
Top wall	1	0	$\partial p/\partial y = 0$
Bottom wall	0	0	$\partial p/\partial y = 0$
Left wall	0	0	$\partial p/\partial x = 0$
Right wall	0	0	$\partial p/\partial x = 0$

Table 27. Driven Cavity: Boundary conditions

Reynolds number is defined as:

$$Re = \frac{\rho u_{ref} L}{\mu} \qquad \textit{(Eq. 105)}$$

and Reynolds numbers of 100, 400, 1000, 5000, 10000 will be studied.

3.2.1.3. Discretization

The discretization of the differential equations required for the FSM are based in the conditions and hypothesis made in *Chapter 3.1.* First of all, a structured-staggered mesh will be used as was described in *Chapter 3.1.2*. *Figure 34* shows the mesh scheme for a 2D case applying FSM and the following equation-discretization for this case of study will be based on that scheme.

The equation-discretization will be done following the next order that describe FSM[11]:

- $\boldsymbol{R}(\vec{\upsilon})$ *Eq. 95*
- Intermediate velocity $\vec{\upsilon}^{p}$ *Eq. 100*
- Poisson equation Δp^{n+1} *Eq. 102*
- Real velocity field $\vec{\upsilon}^{n+1}$ *Eq. 103*

First, the notation must be clarified. The 2D velocity field is given by $\vec{u} = u\hat{\imath} + v\hat{\jmath}$ therefore, u will be sued for the horizontal velocity and v for the vertical velocity.

Having defined the notation, the first step is to discretize $\boldsymbol{R}(\vec{u})$ in a 2D case, and it is obtained:

$$\boldsymbol{R}(u) = \mu(\Delta u) - (\rho\vec{\upsilon}\cdot\nabla)u \qquad \textit{(Eq. 106a)}$$

$$\boldsymbol{R}(v) = \mu(\Delta v) - (\rho\vec{\upsilon}\cdot\nabla)v \qquad \textit{(Eq. 106b)}$$

Integrating over the staggered meshes in X and Y control volumes:

$$\int_{\boldsymbol{\Omega_x}} \boldsymbol{R}(u)\, d\Omega_x = \int_{\boldsymbol{\Omega_x}} \mu(\Delta u)\, d\Omega_x - \int_{\boldsymbol{\Omega_x}} (\rho\vec{\upsilon}\cdot\nabla)u\, d\Omega_x \qquad \textit{(Eq. 107a)}$$

$$\int_{\boldsymbol{\Omega_y}} \boldsymbol{R}(v)\, d\Omega_y = \int_{\boldsymbol{\Omega_y}} \mu(\Delta v)\, d\Omega_y - \int_{\boldsymbol{\Omega_y}} (\rho\vec{\upsilon}\cdot\nabla)v\, d\Omega_y \qquad \textit{(Eq. 107b)}$$

Applying Gauss theorem:

$$\int_{\boldsymbol{\Omega_x}} \boldsymbol{R}(u)\, d\Omega_x = \int_{\boldsymbol{\partial\Omega_x}} \mu\nabla u\cdot\boldsymbol{n}\, dS - \int_{\boldsymbol{\partial\Omega_x}} (\rho\vec{\upsilon})u\cdot\boldsymbol{n}\, dS \qquad \textit{(Eq. 108a)}$$

$$\int_{\boldsymbol{\Omega_y}} \boldsymbol{R}(v)\, d\Omega_y = \int_{\boldsymbol{\partial\Omega_y}} \mu\nabla v\cdot\boldsymbol{n}\, dS - \int_{\boldsymbol{\partial\Omega_y}} (\rho\vec{\upsilon})v\cdot\boldsymbol{n}\, dS \qquad \textit{(Eq. 108b)}$$

Consequently, the discretization:

$$\boldsymbol{R}(u)d\Omega_x = \left[\mu_e \frac{u_E - u_P}{\delta_{EP}}\Delta y - \mu_w \frac{u_P - u_W}{\delta_{WP}}\Delta y + \mu_n \frac{u_N - u_P}{\delta_{NP}}\Delta x - \mu_s \frac{u_P - u_S}{\delta_{SP}}\Delta x\right] - \left[(\rho u)_e u_e \Delta y - (\rho u)_w u_w \Delta y + (\rho v)_n u_n \Delta x - (\rho v)_s u_s \Delta x\right] \quad \textit{(Eq. 109a)}$$

$$\boldsymbol{R}(v)d\Omega_y = \left[\mu_e \frac{v_E - v_P}{\delta_{EP}}\Delta y - \mu_w \frac{v_P - v_W}{\delta_{WP}}\Delta y + \mu_n \frac{v_N - v_P}{\delta_{NP}}\Delta x - \mu_s \frac{v_P - v_S}{\delta_{SP}}\Delta x\right] - \left[(\rho u)_e v_e \Delta y - (\rho u)_w v_w \Delta y + (\rho v)_n v_n \Delta x - (\rho v)_s v_s \Delta x\right] \quad \textit{(Eq. 109b)}$$

It could be seen in *Eq. 109 a, 109b* that the convective terms $((\rho u)_e, (\rho u)_w, (\rho u)_n, (\rho u)_s)$ need to be defined, for which *Figure 36* explains the scheme of the mass flows in the control volumes.

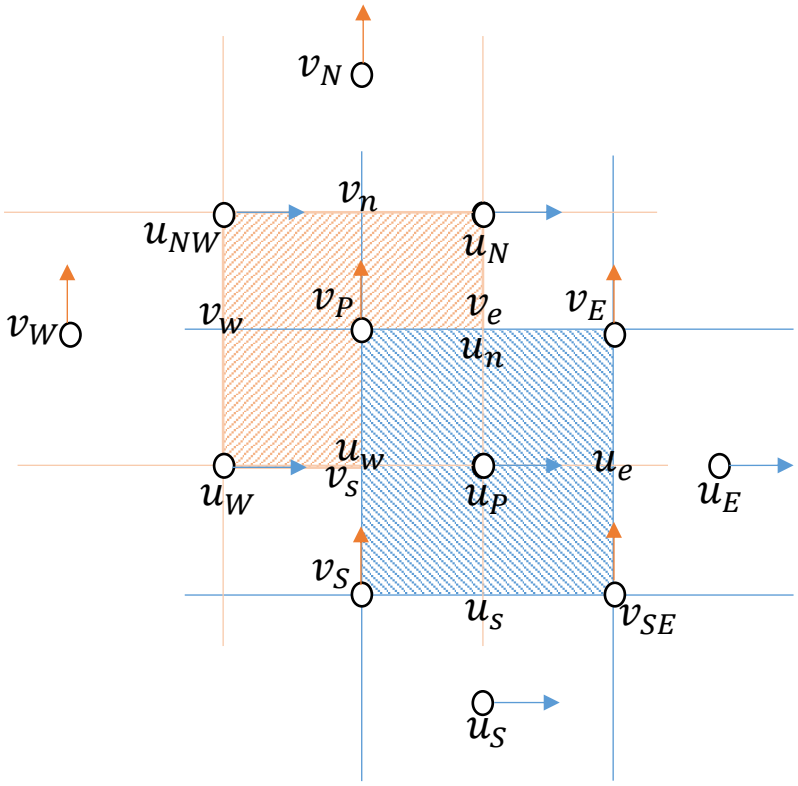


Figure 36. Driven Cavity: Mass flow scheme for a 2D staggered mesh

In that way, the mass flows *Eq. 110* will be computed using CDS (Central Difference Scheme. See Chapter *2.2*)

$$(\rho u)_e = \frac{(\rho u)_E + (\rho u)_P}{2} \quad \textit{(Eq. 110a)}$$

$$(\rho u)_w = \frac{(\rho u)_W + (\rho u)_P}{2} \qquad \textit{(Eq. 110b)}$$

$$(\rho v)_n = \frac{(\rho v)_E + (\rho v)_P}{2} \qquad \textit{(Eq. 110c)}$$

$$(\rho v)_s = \frac{(\rho v)_S + (\rho v)_{SE}}{2} \qquad \textit{(Eq. 110d)}$$

And the values of $u_e, u_w, u_n, u_s, v_e, v_w, v_n, v_s$ will be calculated using numerical schemes (CDS, Upwind, Fromm, QUICK, SMART, See *Chapter 2.2*)

As explained in *Chapter 2.2.2* for High-Resolution schemes the deferred correction must be integrated to the *Eq. 109a, 109b*; therefore, the Equation for high-resolution is defined:

$$\begin{aligned}\boldsymbol{R}(u)d\Omega_x = \Big[&\mu_e \frac{u_E - u_P}{\delta_{EP}}\Delta y - \mu_w \frac{u_P - u_W}{\delta_{WP}}\Delta y + \mu_n \frac{u_N - u_P}{\delta_{NP}}\Delta x \\ &- \mu_s \frac{u_P - u_S}{\delta_{SP}}\Delta x\Big] \\ &- [(\rho u)_e u_e \Delta y - (\rho u)_w u_w \Delta y + (\rho v)_n u_n \Delta x - (\rho v)_s u_s \Delta x] \\ &+ [Fe(u_e^{HRS} - u_e^U) - Fw(u_w^{HRS} - u_w^U) + Fn(u_n^{HRS} - u_n^U) \\ &- Fs(u_s^{HRS} - u_s^U)]\end{aligned} \qquad \textit{(Eq. 111a)}$$

$$\begin{aligned}\boldsymbol{R}(v)d\Omega_y = \Big[&\mu_e \frac{v_E - v_P}{\delta_{EP}}\Delta y - \mu_w \frac{v_P - v_W}{\delta_{WP}}\Delta y + \mu_n \frac{v_N - v_P}{\delta_{NP}}\Delta x \\ &- \mu_s \frac{v_P - v_S}{\delta_{SP}}\Delta x\Big] \\ &- [(\rho u)_e v_e \Delta y - (\rho u)_w v_w \Delta y + (\rho v)_n v_n \Delta x - (\rho v)_s v_s \Delta x] \\ &+ [Fe(v_e^{HRS} - v_e^U) - Fw(v_w^{HRS} - v_w^U) + Fn(v_n^{HRS} - v_n^U) \\ &- Fs(v_s^{HRS} - v_s^U)]\end{aligned} \qquad \textit{(Eq. 111b)}$$

Thus, *Eq. 109a, 109b* are used for Low-Resolution schemes, and *Eq. 111a, 111b* for High-resolution schemes.

Once the vector $\boldsymbol{R}(\vec{u})$ is defined, the second step is to calculate the intermediate velocity.

$$u^p = u^n + \frac{\Delta t}{\rho}\left[\frac{3}{2}\boldsymbol{R}(u)^n - \frac{1}{2}\boldsymbol{R}(u)^{n-1}\right] \qquad \textit{(Eq. 112a)}$$

$$v^p = v^n + \frac{\Delta t}{\rho}\left[\frac{3}{2}\boldsymbol{R}(v)^n - \frac{1}{2}\boldsymbol{R}(v)^{n-1}\right] \qquad \textit{(Eq. 112b)}$$

Having calculated the intermediate velocity, the third step is to discretize the Poisson equation:

$$\int_\Omega \Delta p^{n+1} d\Omega = \frac{\rho}{\Delta t}\int_\Omega \nabla \cdot \vec{v}^p d\Omega \qquad \textit{(Eq. 113)}$$

Applying Gauss theorem: *(Eq. 114)*

$$\int_{\partial\Omega} \Delta p^{n+1} \cdot \boldsymbol{n} dS = \frac{\rho}{\Delta t} \int_{\partial\Omega} \vec{v}^{p} \cdot \boldsymbol{n} dS$$

Discretizing it is obtained:

$$\frac{p_E^{n+1} - p_P^{n+1}}{\delta_{EP}} \Delta y - \frac{p_P^{n+1} - p_W^{n+1}}{\delta_{WP}} \Delta y + \frac{p_N^{n+1} - p_P^{n+1}}{\delta_{NP}} \Delta x - \frac{p_P^{n+1} - p_S^{n+1}}{\delta_{SP}} \Delta x = \frac{1}{\Delta t} [(\rho u^P)_e \Delta y - (\rho u^P)_w \Delta y + (\rho v^P)_n \Delta x - (\rho v^P)_s \Delta x] \quad \textit{(Eq. 115)}$$

In order to solve *Eq. 115*, the coefficients for the algebraic solution *Eq. 116* are presented in *Table 28 and 29*.

$$a_P P_P = a_E P_E + a_W P_W + a_N P_N + a_S P_S + b \quad \textit{(Eq. 116)}$$

Coefficient	Value for middle nodes (Pressure Field)	
a_E	$a_E = \frac{\Delta y}{(\delta x)_{PE}}$	*(Eq. 117a)*
a_W	$a_W = \frac{\Delta y}{(\delta x)_{PW}}$	*(Eq. 117b)*
a_N	$a_N = \frac{\Delta x}{(\delta y)_{PN}}$	*(Eq. 117c)*
a_S	$a_S = \frac{\Delta x}{(\delta y)_{PS}}$	*(Eq. 117d)*
a_P	$a_P = a_E + a_W + a_N + a_S$	*(Eq. 117e)*
b	$b = -\frac{1}{\Delta t}[(\rho u^P)_e \Delta y - (\rho u^P)_w \Delta y + (\rho v^P)_n \Delta x - (\rho v^P)_s \Delta x]$	*(Eq. 117f)*

Table 28. Driven Cavity: Coefficients middle nodes to solve pressure field in the Poisson equation

Coefficient	Values of boundary conditions (Pressure Field)							
	West Wall	East Wall	South Wall	North Wall	West North	East North	West South	East South
a_E	1	0	0	0	0.5	0	0.5	0
a_W	0	1	0	0	0	0.5	0	0.5
a_N	0	0	1	0	0	0	0.5	0.5
a_S	0	0	0	1	0.5	0.5	0	0
a_P	1	1	1	1	1	1	1	1
b	0	0	0	0	0	0	0	0

Table 29. Driven Cavity: Boundary coefficients to solve pressure field in the Poisson Equation

Finally, the las step is to calculate the real velocity with the pressure field found.

$$u^{n+1} = u^{p} - \frac{\Delta t}{\rho}\left(\frac{\partial p}{\partial x}\right)^{n+1} \quad \textit{(Eq. 118a)}$$

$$v^{n+1} = v^{p} - \frac{\Delta t}{\rho}\left(\frac{\partial p}{\partial y}\right)^{n+1} \quad \textit{(Eq. 118b)}$$

Discretizing:

$$u^{n+1} = u^{p} - \frac{\Delta t}{\rho}\left[\frac{p_e^{n+1} - p_w^{n+1}}{\Delta x}\right] \quad \textit{(Eq. 119a)}$$

$$v^{n+1} = v^{p} - \frac{\Delta t}{\rho}\left[\frac{p_n^{n+1} - p_s^{n+1}}{\Delta y}\right] \quad \textit{(Eq. 119b)}$$

Due to the fact that it has been used a fully explicit scheme, the time-step parameter is a crucial value that could intervene in the stability of the numerical resolution. Is for that reason that a CFL(Courant-Friedrich-Levy) condition must be defined and control this parameter.

CFL define a minimum time-step necessary for stability for convective *Eq. 120a* and diffusive *Eq. 120b* terms:

$$\Delta t_c = \min\left(0.35\frac{\Delta x}{|\vec{\upsilon}|}\right) \quad \textit{(Eq. 120a)}$$

$$\Delta t_d = \min\left(0.20\frac{\rho(\Delta x)^2}{\mu}\right) \quad \textit{(Eq. 120b)}$$

The optimal time-step will be the minimum of the convective-diffusive terms:

$$\Delta t = \min(\Delta t_c, \Delta t_d) \quad \textit{(Eq. 121)}$$

3.2.1.4. Algorithm

With the purpose of graphically explain the code done for this case, *Figure 37* shows the algorithm flow chart that briefly explain the main processes inside the code. The code consists of one main code, which is the one that is used to manually enter parameters and run the code, and five more sub-codes: The mesh construction, computation of discretization coefficients, sub-code to find $\boldsymbol{R}(\vec{\upsilon})$, to solve Poisson equation, and finally a sub-code for numerical solvers. See *Attachment I – 2.2* for the algorithm implemented.

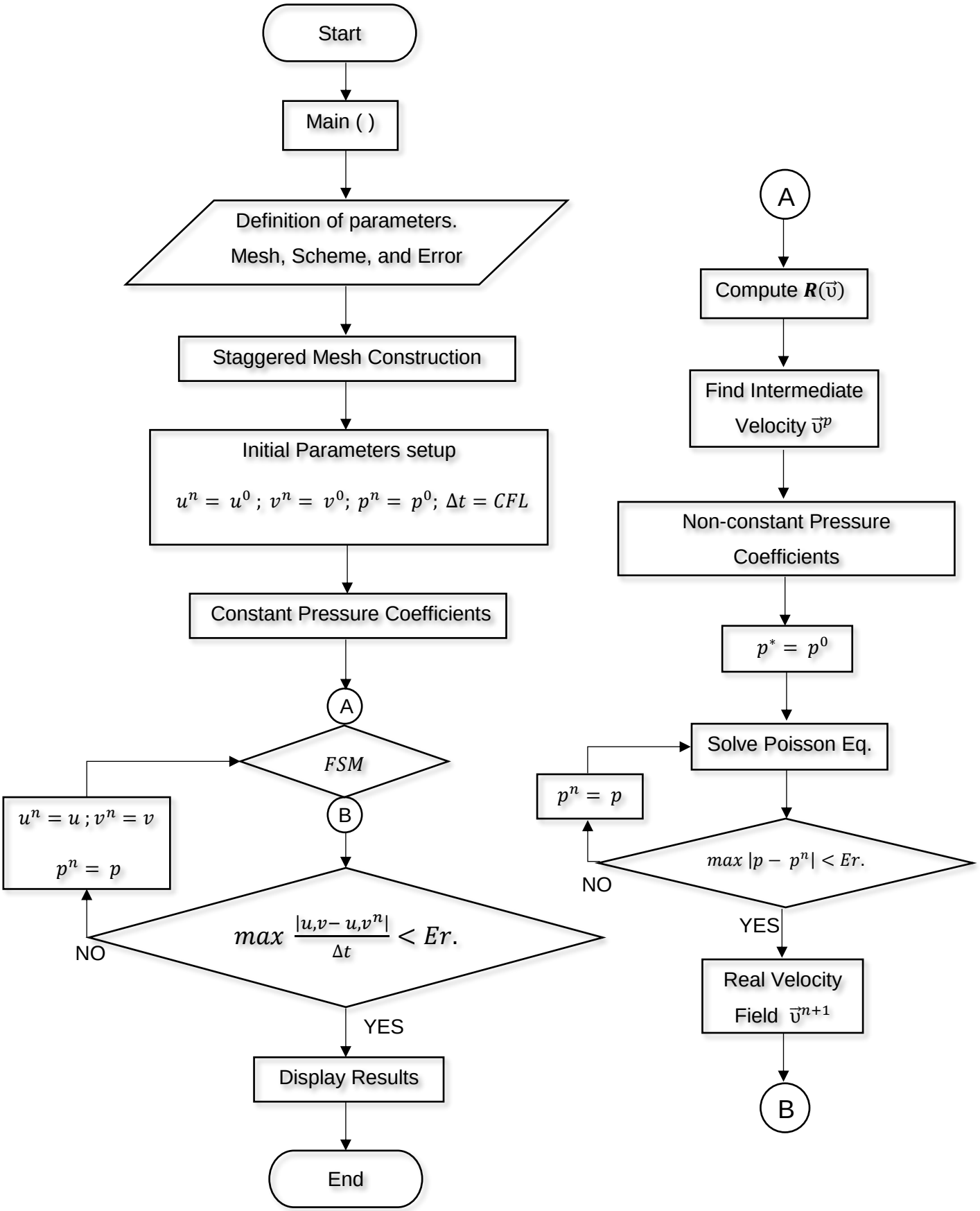


Figure 37. Driven Cavity: Algorithm flow chart

3.2.1.5. Results

In order to validate the results for this case and compare them with a scientific reference, the document *"High-Re Solutions for incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method*"[12] gives a reference table with results for the velocity field in the case of the Driven cavity. *Tables 30* and *31* contains the reference data that will be compared with the results obtained by the code generated.

Furthermore, the results will be analyzed in three parameters as priory done. First a study of the mesh-size effect on the results ([29x29], [71x71], [101x101], [129x129]), the numerical scheme implemented (Upwind, QUICK, SMART, Fromm), finally an analysis of the Reynold number of stability and accuracy (Re = 100, 400,1000,5000,10000).

Ux (Horizontal velocity along vertical line at geometric center)							
Y	**Reynolds**						
	100	**400**	**1000**	**3200**	**5000**	**7500**	**10000**
1	1	1	1	1	1	1	1
0.9766	0.84123	0.75837	0.65928	0.53236	0.48223	0.47244	0.47221
0.9688	0.78871	0.68439	0.57492	0.48296	0.4612	0.47048	0.47783
0.9609	0.73722	0.61756	0.51117	0.46547	0.45992	0.47323	0.4807
0.9531	0.68717	0.55892	0.46604	0.46101	0.46036	0.47167	0.47804
0.8516	0.23151	0.29093	0.33304	0.34682	0.33556	0.34228	0.34635
0.7344	0.00332	0.16256	0.18719	0.19791	0.20087	0.20591	0.20673
0.6172	-0.13641	0.02135	0.05702	0.07156	0.08183	0.08342	0.08344
0.5	-0.20581	-0.11477	-0.0608	-0.04272	-0.03039	-0.038	-0.03111
0.4531	-0.2109	-0.17119	-0.10648	-0.86636	-0.07404	-0.07503	-0.0754
0.2813	-0.15662	-0.32726	-0.27805	-0.24427	-0.22855	-0.23176	-0.23186
0.1719	-0.1015	-0.24299	-0.38289	-0.34323	-0.3305	-0.32393	-0.32709
0.1016	-0.06434	-0.14612	-0.2973	-0.41932	-0.40435	-0.38324	-0.38
0.0703	-0.04775	-0.10338	-0.2222	-0.37827	-0.43643	-0.43025	-0.41657
0.0625	-0.04192	-0.09266	-0.20196	-0.35344	-0.42901	-0.4359	-0.42537
0.0547	-0.03717	-0.08186	-0.18109	-0.32407	-0.41165	-0.43154	-0.42735
0	0	0	0	0	0	0	0

***Table 30. Driven Cavity: Reference Ux velocity along vertical line at geometric center*[12]**

Vy (Vertical velocity along horizontal line at geometric center)							
X	Reynolds						
	100	400	1000	3200	5000	7500	10000
1	0	0	0	0	0	0	0
0.9688	-0.05906	-0.12146	-0.21388	-0.39017	-0.49774	-0.53858	-0.54302
0.9609	-0.07391	-0.15663	-0.27669	-0.47425	-0.55069	-0.55216	-0.52987
0.9531	-0.08864	-0.19254	-0.33714	-0.52357	-0.55408	-0.52347	-0.49099
0.9453	-0.10313	-0.22847	-0.39188	-0.54053	-0.52876	-0.4889	-0.45863
0.9063	-0.16914	-0.23827	-0.5155	-0.44307	-0.41442	-0.4105	-0.41496
0.8594	-0.22445	-0.44993	-0.42665	-0.37401	-0.36214	-0.36213	-0.36737
0.8047	-0.24533	-0.38598	-0.31966	-0.31184	-0.30018	-0.30448	-0.30719
0.5	0.05454	0.05186	0.02526	0.00999	0.00945	0.00824	0.00831
0.2344	0.17527	0.30174	0.32235	0.28188	0.2728	0.27348	0.27224
0.2266	0.17507	0.30203	0.33075	0.2903	0.28066	0.28117	0.28003
0.1563	0.16077	0.28124	0.37095	0.37119	0.35368	0.3506	0.3507
0.0938	0.12317	0.22965	0.32627	0.42768	0.42951	0.41824	0.41487
0.0781	0.1089	0.2092	0.30353	0.41906	0.43648	0.43564	0.43124
0.0703	0.10091	0.19713	0.29012	0.40917	0.43329	0.4403	0.43733
0.0625	0.09233	0.1836	0.27485	0.3956	0.42447	0.43979	0.43983
0	0	0	0	0	0	0	0

Table 31. Driven Cavity: Reference Vy velocity along horizontal line at geometric center[12]

- **Mesh (at Re = 1000, Numerical scheme: QUICK)**

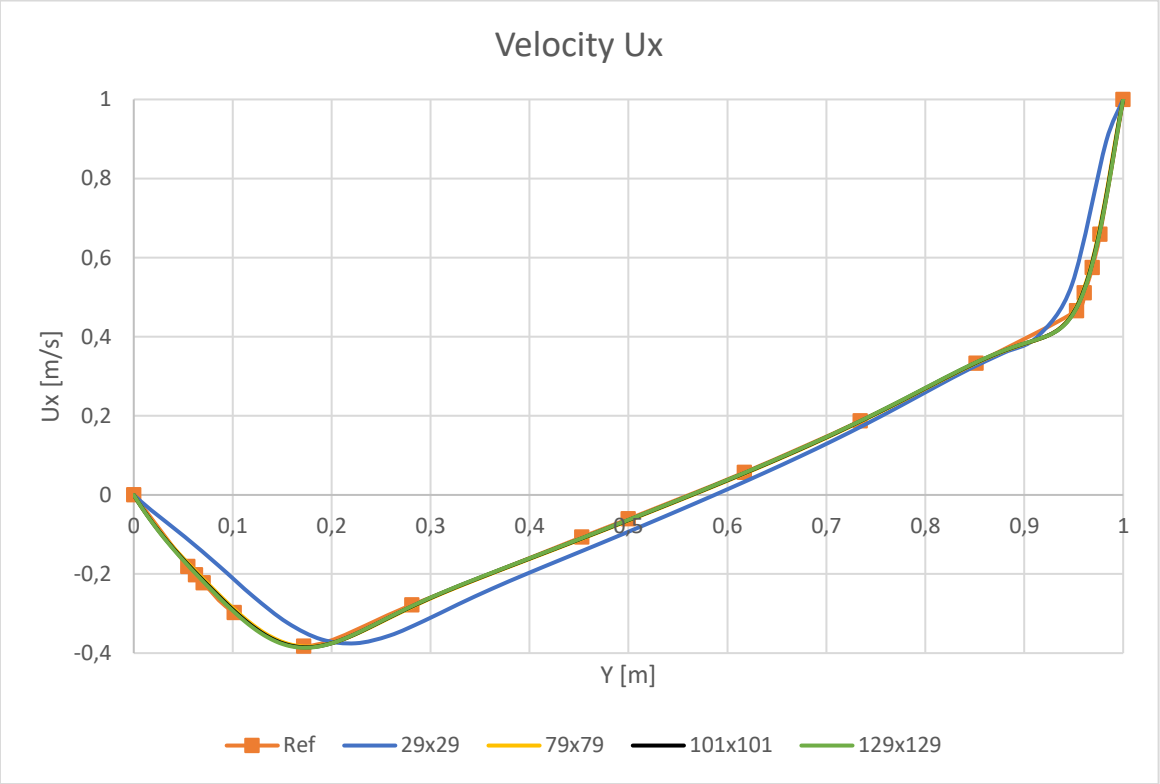


Figure 38. Driven cavity: Mesh comparison for Ux velocity

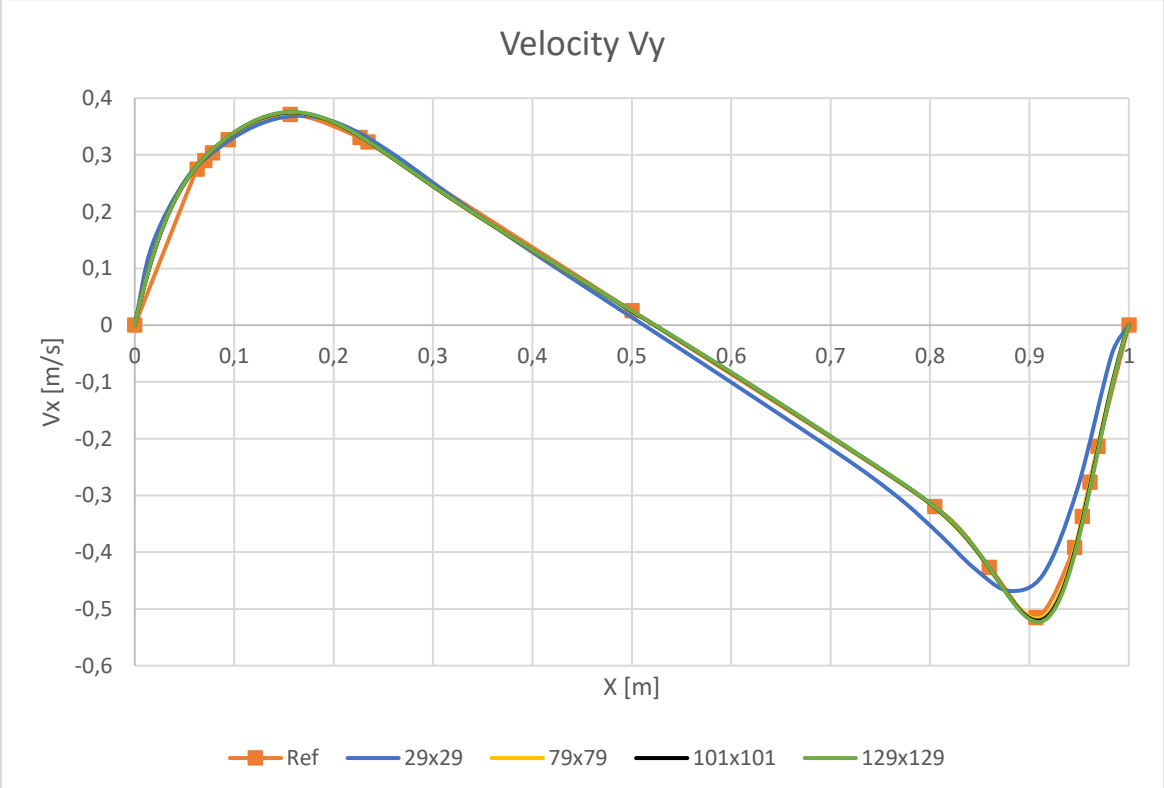


Figure 39. Driven Cavity: Mesh comparison for Vy velocity

- **Scheme (at Re = 1000, Mesh size [101x101])**

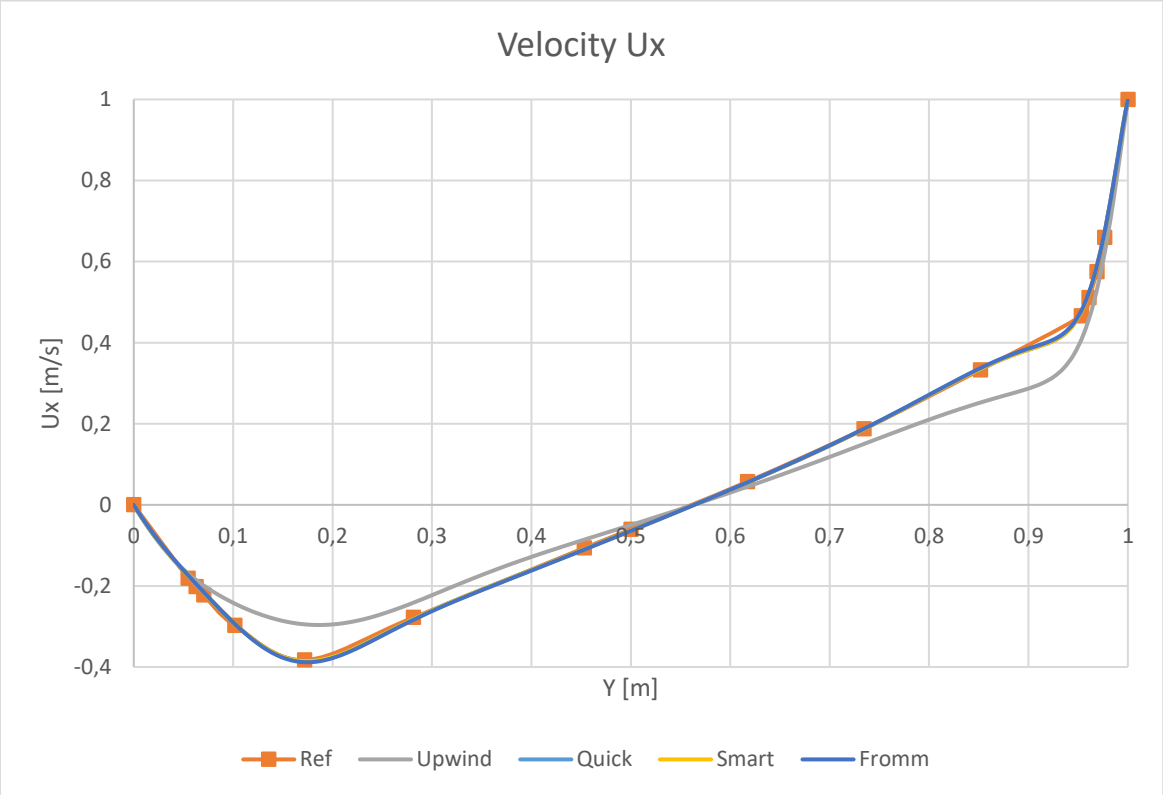


Figure 40. Driven Cavity: Numerical scheme comparison for Ux velocity

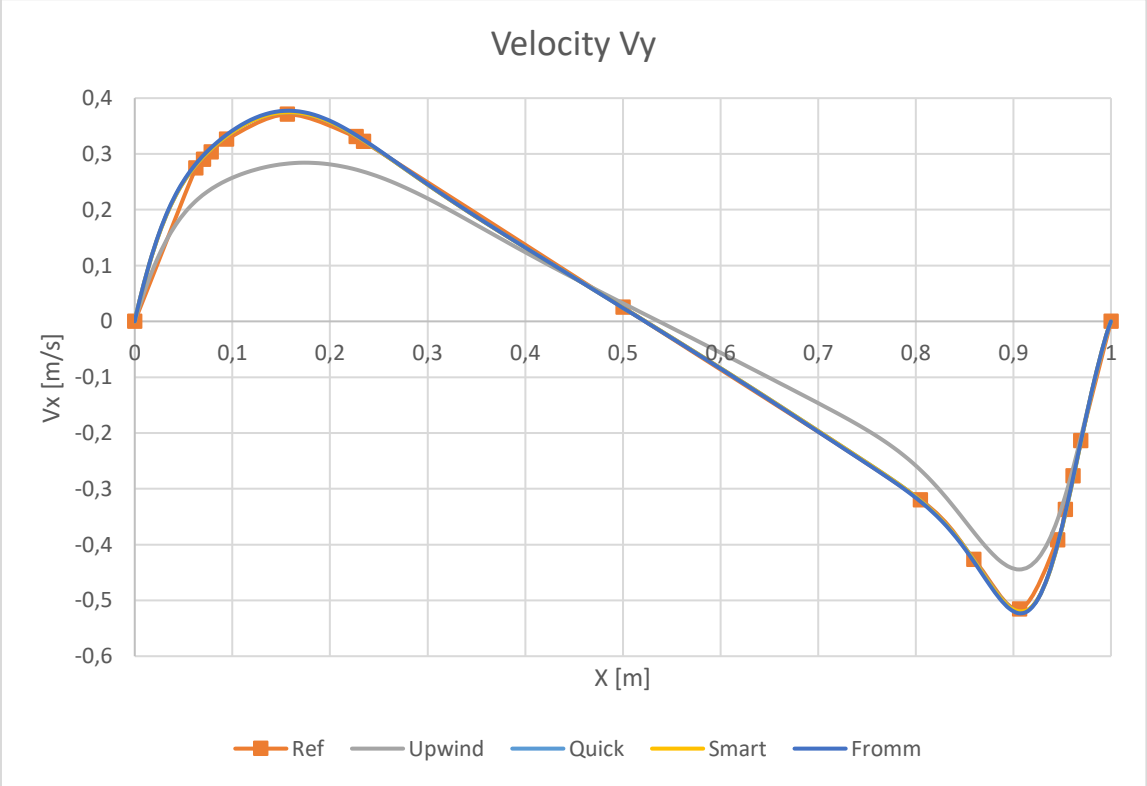


Figure 41. Driven Cavity: Numerical scheme comparison for Vy velocity

- **Reynolds Number (Mesh size [129x129], Numerical scheme: QUICK)**

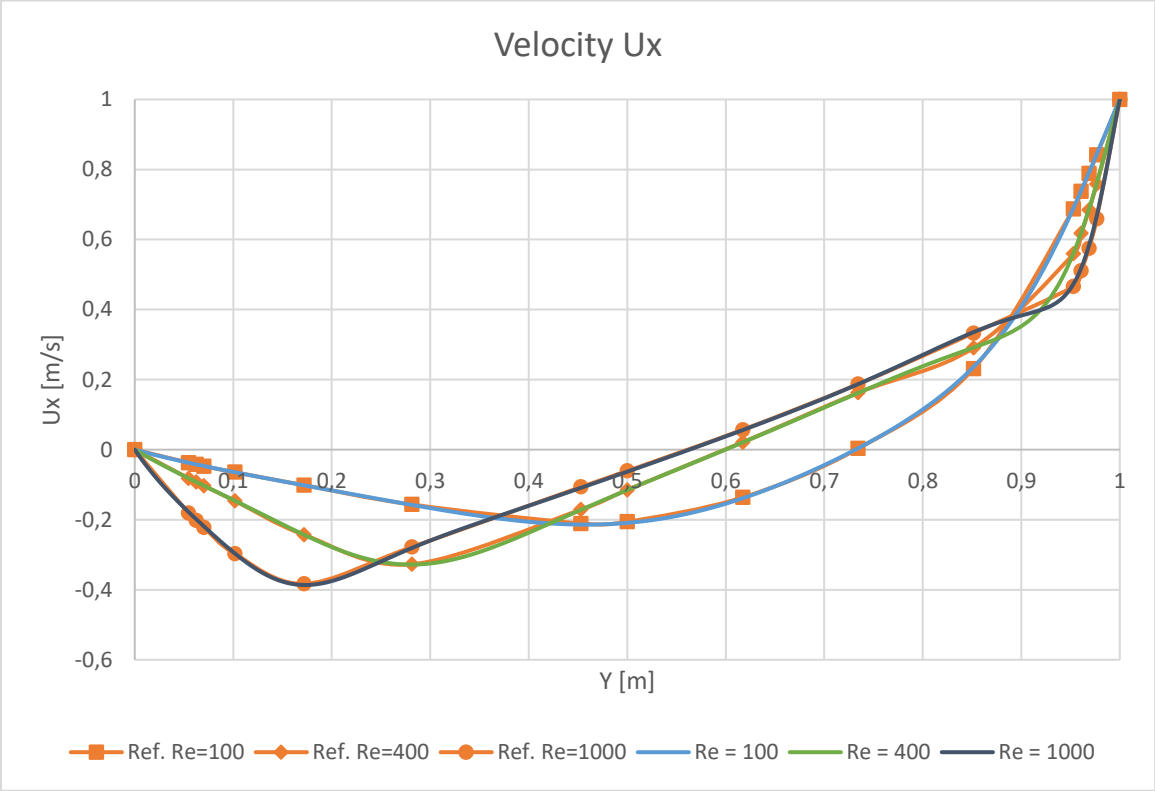


Figure 42. Driven Cavity: Low-Reynolds number comparison for Ux velocity

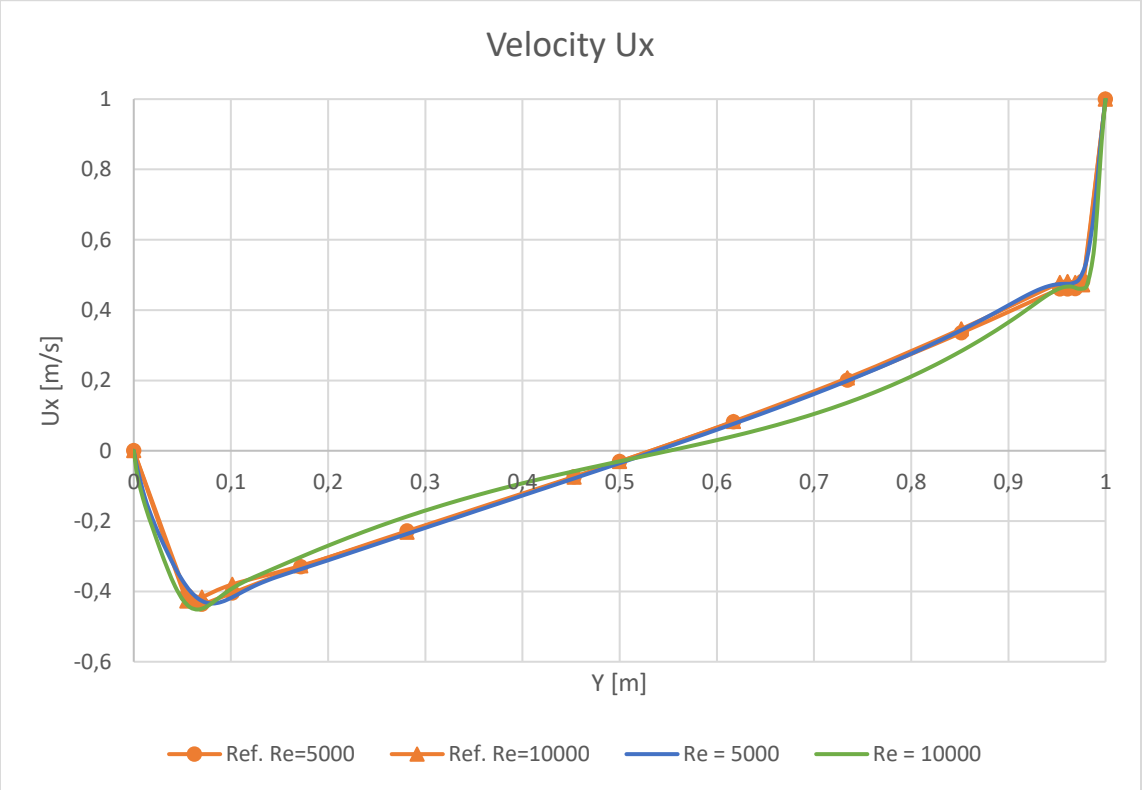


Figure 43. Driven Cavity: High-Reynolds number comparison for Ux velocity

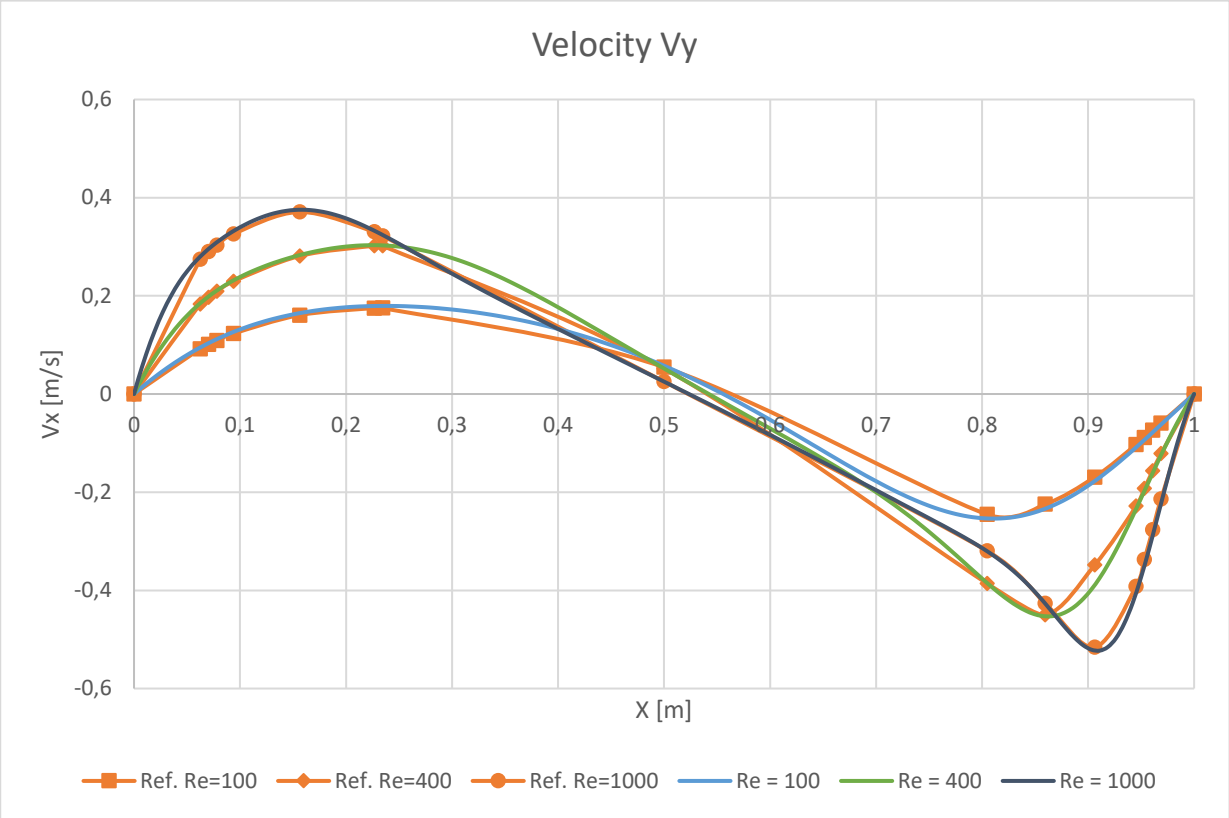


Figure 44. Driven Cavity: Low-Reynolds number comparison for Vy velocity

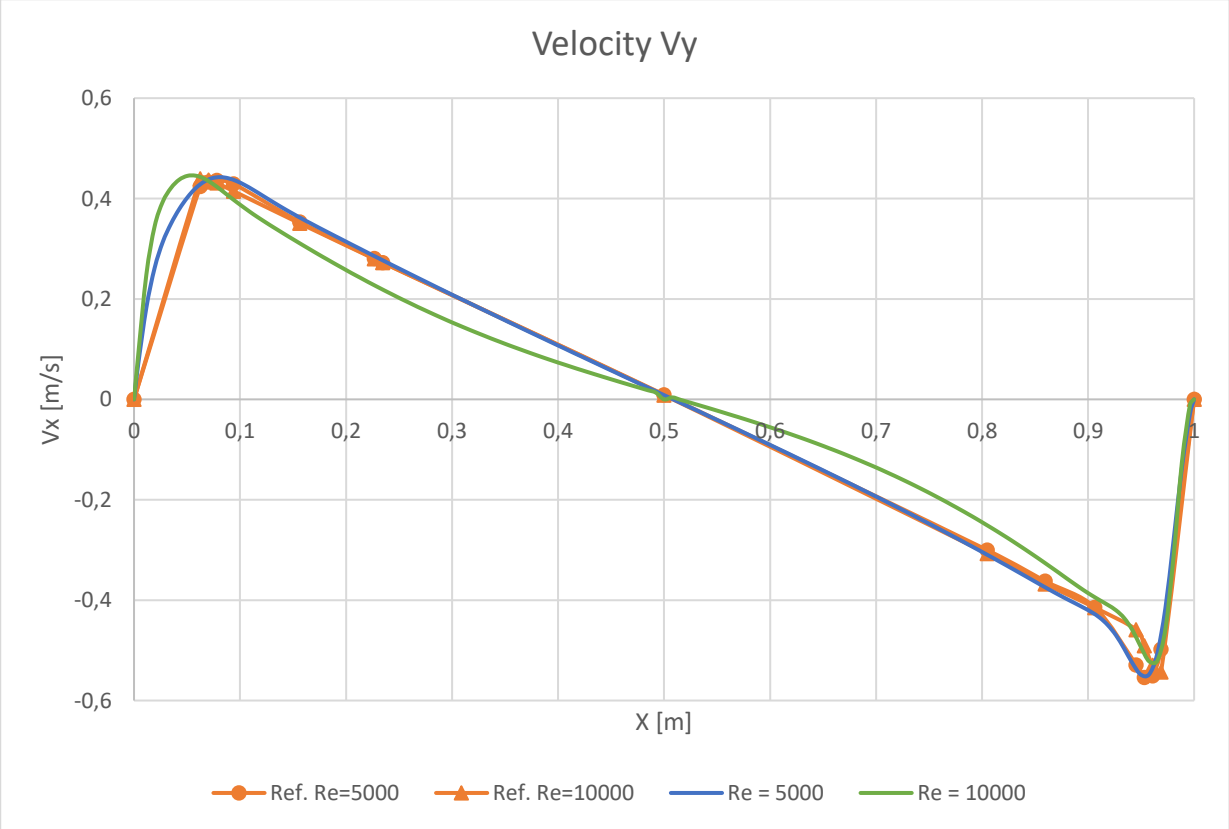


Figure 45. Driven Cavity: High-Reynolds number comparison for Vy velocity

3.2.1.6. Discussion of the results

In conclusion, the influence of three parameters in the driven cavity case (mesh size, scheme, and Reynolds number) was analyzed and are presented in the following three main conclusions:

- The size of the mesh does affect the solution by incrementing the diffusive effect for coarse meshes, However, this effect is more accentuated for the coarsest mesh [29x29]. The steep on the [29x29] curve is always greater than the rest of the mesh sizes. this is a result of the diffusive effect produced by the coarsest meshes. In *Figures 38 and 39* it could be seen how the finest meshes follow very closely the reference curve. Another parameter that need to be considered is that mesh size of [129x129] took almost 50% more time than [79x79], but the accuracy of both mesh sizes is close, so it must be analyzed the convenience of using finer meshes despite the computational cost.
- The results showed that the accuracy of the schemes applied to this case is quite similar and accurate enough. However, High-resolution scheme are more accurate than the low-order scheme used (Upwind). analyzing other parameters, such as time, iteration and convergence, the high-order schemes seem to last more and use more computational memory. High-order schemes used in average 8500 seconds to converge, while the low-order scheme Upwind used 80% less time than the High-order schemes. Therefore, upwind scheme is more efficient because the computational cost, but it is less accurate than high-resolution schemes as could be seen in *Figures 40* and *41*. The steep of the upwind curve is lesser than the high-order schemes. Among, High-order schemes the accuracy is similar with differences of less than 3%; however, QUICK used less time and computational resources than SMART and Fromm schemes.
- The results for the variation of the Reynolds number follow the reference curve given. Nonetheless, for Re = 10000 the code did not reach the convergence criteria because it took a great amount of time. the simulation was done after 20000 iterations with a convergence criterion of $3.4x10^{-3}$. This is because at Re = 10000 the turbulence effect becomes significant and the mesh size is not fine enough to capture the small scales of turbulence. That is the reason of the discrepancy between the refence curve and the results in *Figure 45.* For the rest of the Reynolds number (100, 400, 1000, and 5000) the results are similar to the reference. Another consideration is the time-step used for high Reynolds, where the convective and diffusive coefficients when from 0.35 and 0.2 to 0.2 and 0.08 respectively, in order to reach the convergence criteria.

3.2.2. Flow Around a Square Cylinder

3.2.2.1. Objective

The second case of study can be now applied to verify the effectivity of the algorithm. The problem is proposed by CTTC, and it consist of a 2D aerodynamic analysis of a square cylinder under a horizontal flow. Velocity field, pressure field, and the aerodynamics coefficients (drag, and lift) are requested for different Reynolds numbers.

3.2.2.2. Problem Definition

The case of study proposed consist of the aerodynamic analysis of a horizontal flow around a square cylinder in a plane channel for Reynolds number from 1 up to 200. Reference results are used to validate the data about the aerodynamic coefficients. The geometry parameters and general scheme of the case is show in *Figure 46*:

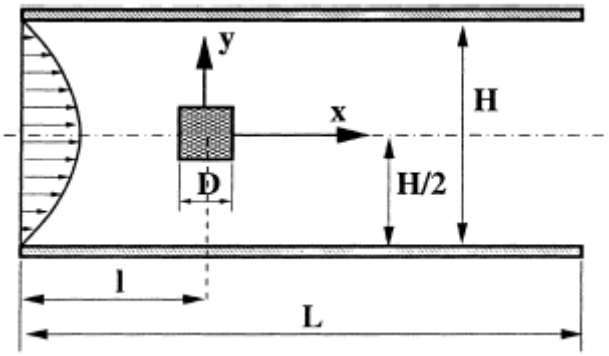


Figure 46. Square Cylinder: Geometry and general scheme (Extracted from* [13]*)

The square cylinder is defined by D which will define the blockage ratio $B = D/H = 1/8$. Furthermore, the length is defined as $L = 50D$ to reduce the influence of the inflow/outflow effects[13]. $l = L/4$ is the inflow length.

The study will be developed for Reynolds numbers 1, 10, 30, 60, 100, and 200. Therefore, the Reynolds number will be defined as:

$$Re = \frac{\rho u_{ref} D}{\mu} \qquad \textit{(Eq. 122)}$$

In order to work in dimensionless form, D, ρ, and u_{ref} will be set to 1. Therefore, the geometric values become: $D = 1,\ H = 8,\ L = 50,\ l = 12.5,\ \rho = 1,\ u_{ref} = 1$

u_{ref} will be the maximum horizontal velocity in the inflow profile throughout the plane channel. The inflow profile is defined by a parabolic function for Y (vertical direction) with

a maximum value of 1 in the horizontal line at the geometric center of the longitude plane of the channel. The inflow parabolic-profile equation and boundary conditions for the channel and square cylinder are defined in *Table 32*.

Wall	Values of boundary conditions		
	Ux (Horizontal Velocity)	Vy (Vertical Velocity)	Pressure
Top wall	0	0	$\frac{\partial p}{\partial y} = 0$
Bottom wall	0	0	$\frac{\partial p}{\partial y} = 0$
Left wall	$\frac{1}{2}y - \frac{1}{16}y^2$	0	$\frac{\partial p}{\partial x} = 0$
Right wall	$\frac{\partial u}{\partial x} = 0$	$\frac{\partial v}{\partial x} = 0$	$p = 0$

Table 32. Square Cylinder: Boundary conditions

It is important to note that the inflow parabolic-profile equation is defined using the left-bottom corner as the origin of the coordinates.

3.2.2.3. Discretization

The discretization process for the fractional step method is the same that the followed in the prior case of study. See *Chapter 3.2.1.3*. however, there are two additional processes that need to be clarify. First of all, the mesh is generated equally distributed throughout the domain. Therefore, there will be nodes which are inside the square cylinder. Those nodes will be set a value of zero for horizontal and vertical velocities, and for pressure. Setting these parameters to zero will represent the presence of the square cylinder obstacle in the algorithm. In the same way, the fluxes neighboring the control volumes inside the square cylinder must be set to zero as well *Figure 47*. The mesh size used was chosen by imposing a fixed number of control volumes within the square cylinder. This number was 20. Therefore, the total mesh size used for this case was [1000x160]. The second additional process is the calculation of the aerodynamic coefficients for a square cylinder for steady and non-steady conditions. Two main parameters will be studied in this case: Lift coefficient C_l and drag coefficient C_d. Lift coefficient is defined as a scalar which models the dependencies of geometry, flow and inclination conditions that affect the lift force[14]. Drag coefficient is defined as a scalar which consider shape, flow conditions as well on the drag force. It expresses the ratio of the drag forces to the force produced by dynamic pressure

force per unit of area[15]. Therefore the drag and lift coefficients con be found using the following formula[16]:

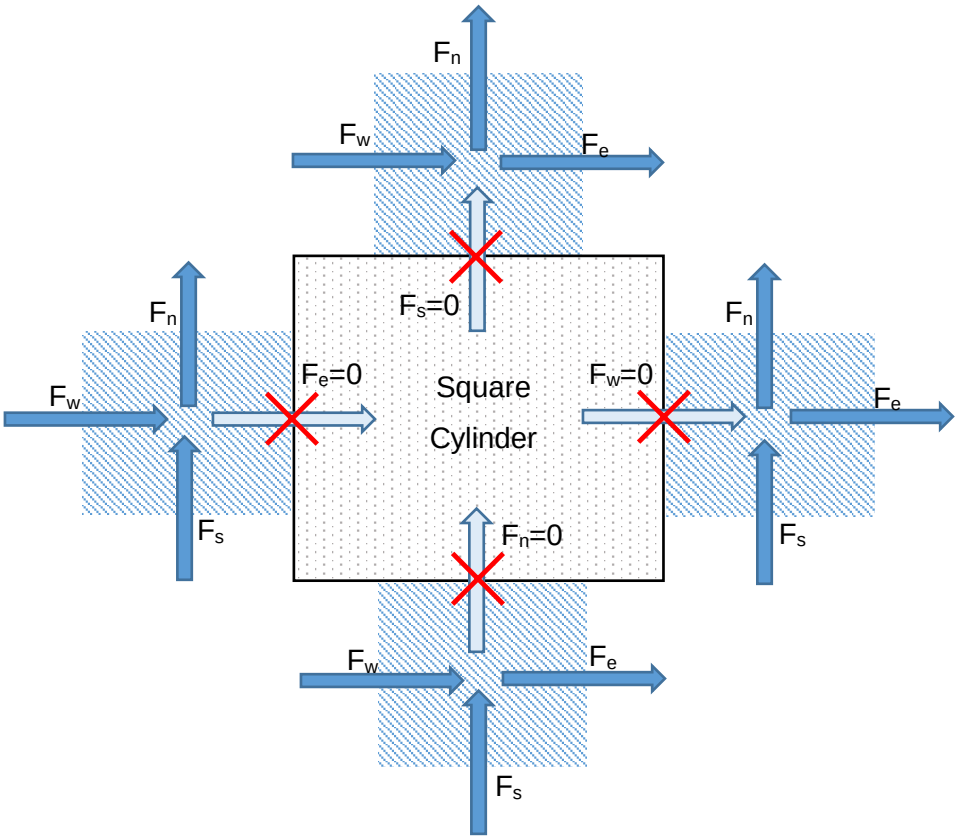


Figure 47. Square Cylinder: Fluxes in the neighbor control volumes to the square cylinder

$$C_l = \frac{2F_l}{\rho u_{max}^2 D} \qquad \textit{(Eq. 123)}$$

$$C_d = \frac{2F_d}{\rho u_{max}^2 D} \qquad \textit{(Eq. 124)}$$

The density ρ, reference velocity u_{max} and diameter D are known and defined before, but to find the aerodynamics forces is necessary analyze the forces over the square cylinder:

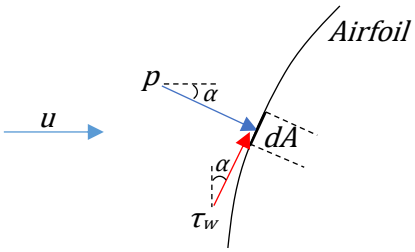


Figure 48. Square Cylinder: Aerodynamic forces on a surface

$$F_l = \int_A dF_y = -\int_A psin(\alpha)dA + \int_A \tau_w \cos(\alpha)dA \qquad \textit{(Eq. 125)}$$

$$F_d = \int_A dF_x = \int_A (pdA)\cos(\alpha) + \int_A (\tau_w dA)\sin(\alpha) \quad \textit{(Eq. 126)}$$

Where the transversal stress τ_w can be found analyzing and discretizing the diffusive term in the Navier-Stokes equation:

$$\nabla \cdot \vec{\tau} = \mu \nabla^2 \vec{\upsilon} \quad \textit{(Eq. 127)}$$

$$\nabla \cdot \vec{\tau} = \mu (\nabla \vec{\upsilon} + (\nabla \vec{\upsilon})^T) \quad \textit{(Eq. 128)}$$

$$\tau = \mu \begin{bmatrix} 2\frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} & 2\frac{\partial v}{\partial y} \end{bmatrix} \quad \textit{(Eq. 129)}$$

Therefore, the tangential stress will be given by

$$\tau_w = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \quad \textit{(Eq. 130)}$$

And discretizing over the pressure nodes:

$$\tau_w = \mu \left(\frac{u_n - u_s}{\Delta y} + \frac{v_e - v_w}{\Delta x} \right) \quad \textit{(Eq. 131)}$$

Applying the integration to the vertical (Left and right) faces of square cylinder it is obtained:

$$F_{l,\uparrow} = \sum \tau_{wi} \Delta y_i \quad \textit{(Eq. 132a)}$$

$$F_{d,\uparrow} = \sum p_i \Delta y_i \quad \textit{(Eq. 132b)}$$

Now to the horizontal faces (top and bottom)

$$F_{l,\rightarrow} = \sum p_i \Delta x_i \quad \textit{(Eq. 133a)}$$

$$F_{d,\rightarrow} = \sum \tau_{wi} \Delta x_i \quad \textit{(Eq. 133b)}$$

And the total aerodynamic forces will be:

$$F_l = F_{l,\uparrow} + F_{l,\rightarrow} \quad \textit{(Eq. 134)}$$

$$F_d = F_{d,\uparrow} + F_{d,\rightarrow} \quad \textit{(Eq. 135)}$$

3.2.2.4. Algorithm

The FSM code used in the prior case is the same implemented for this case; thus, the same *Figure 37* shows the general algorithm flow chart. For this case an extra sub-code was generated to calculate the aerodynamic coefficients. The code consists of one main code, which is the one that is used to manually enter parameters and run the code, and six more sub-codes: The mesh construction, computation of discretization coefficients, sub-code to

find $\boldsymbol{R}(\vec{u})$, to solve Poisson equation, calculation of aerodynamic coefficients, and finally a sub-code for numerical solvers. See *Attachment I – 2.2.* for the algorithm implemented

3.2.2.5. Results

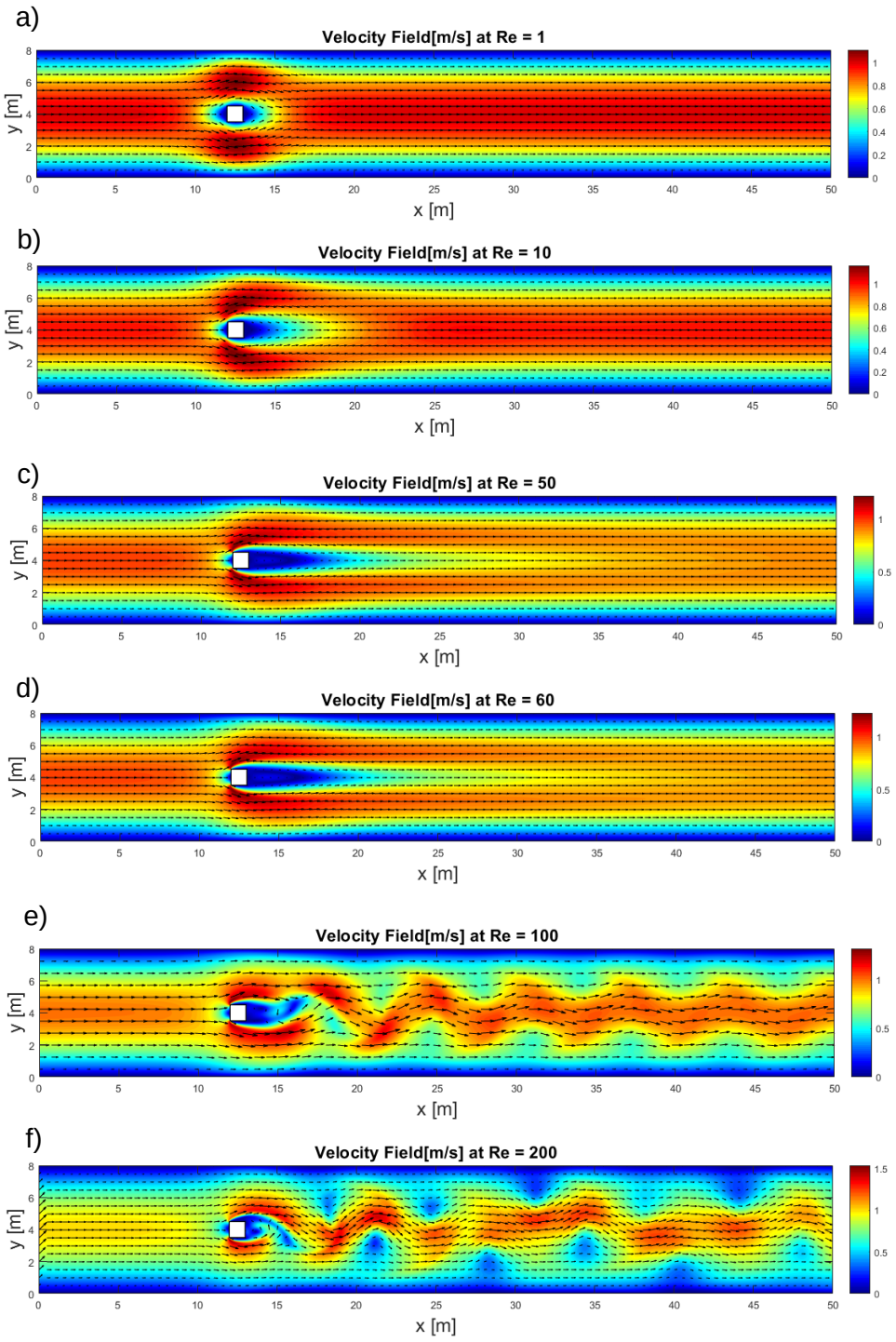


Figure 49. Square Cylinder: Velocity field results at Reynolds a) 1, b) 10, c) 50, d) 60, e) 100, f) 200

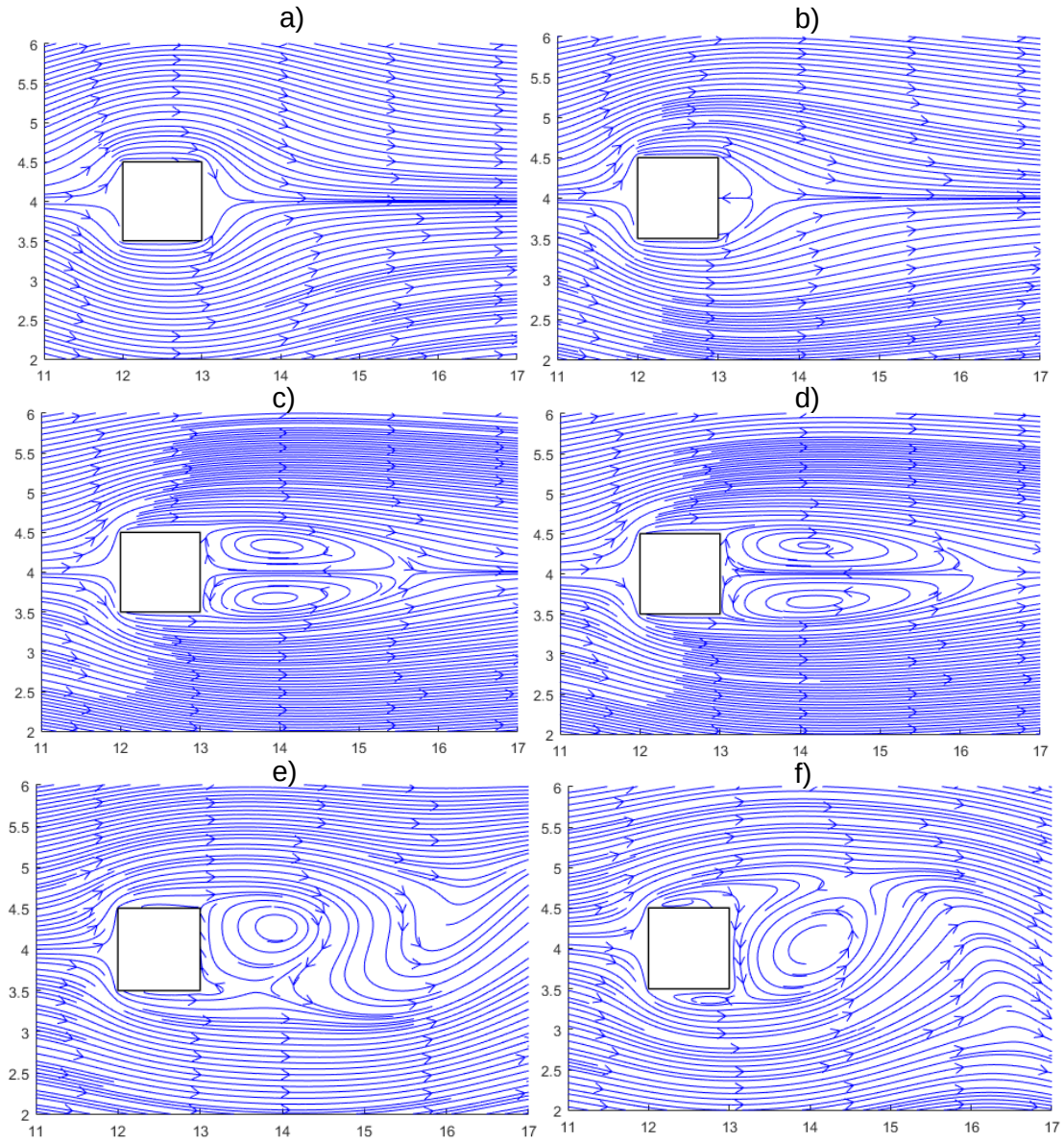


Figure 50. Square Cylinder: Streamlines around the square cylinder at Reynolds a) 1, b) 10, c) 50, d) 60, e) 100, f) 200

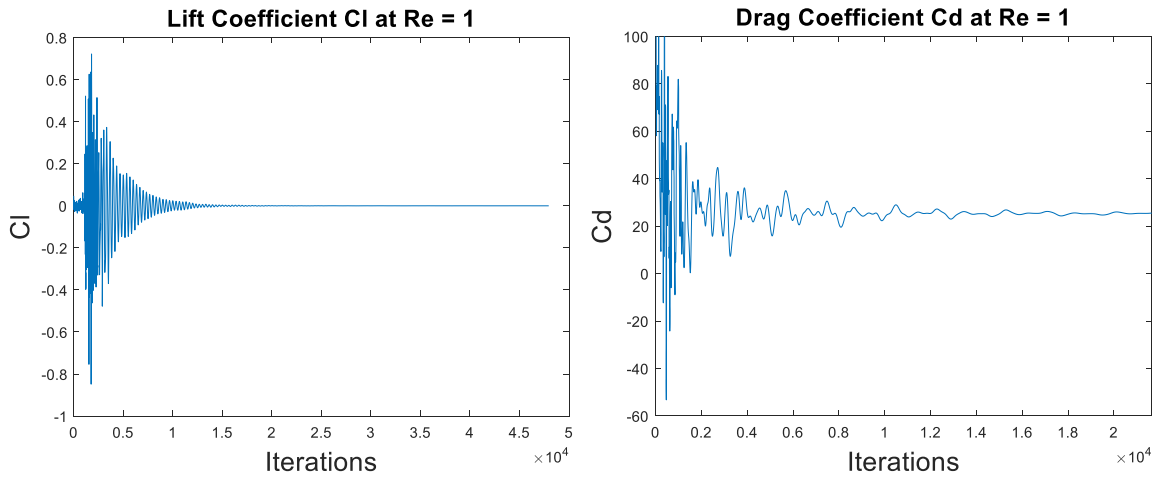
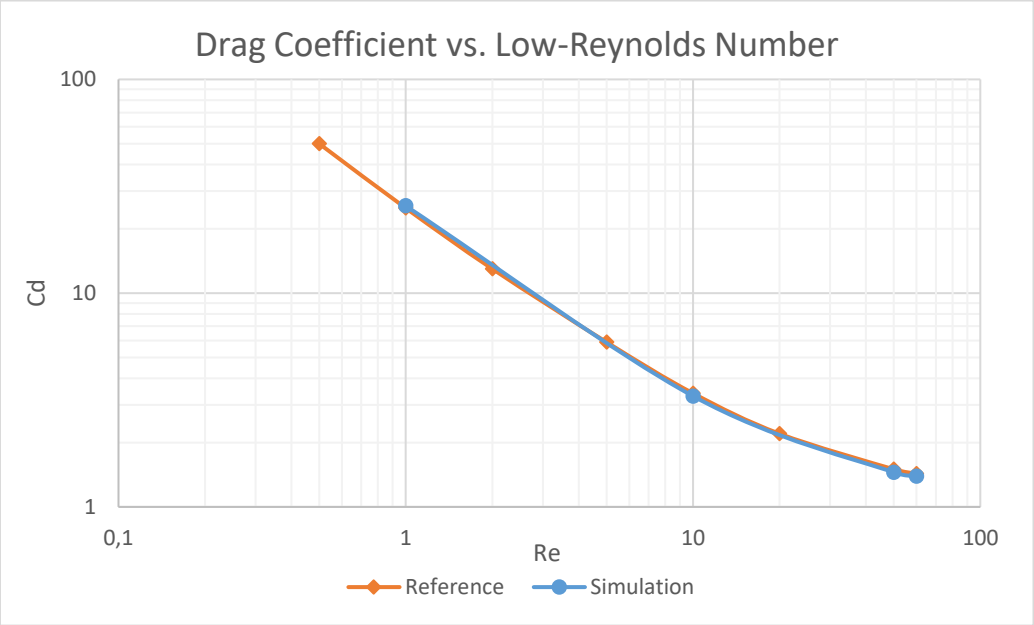


Figure 51. Square Cylinder: Variance of lift and drag coefficient during the simulation at Re = 1

a)



b)

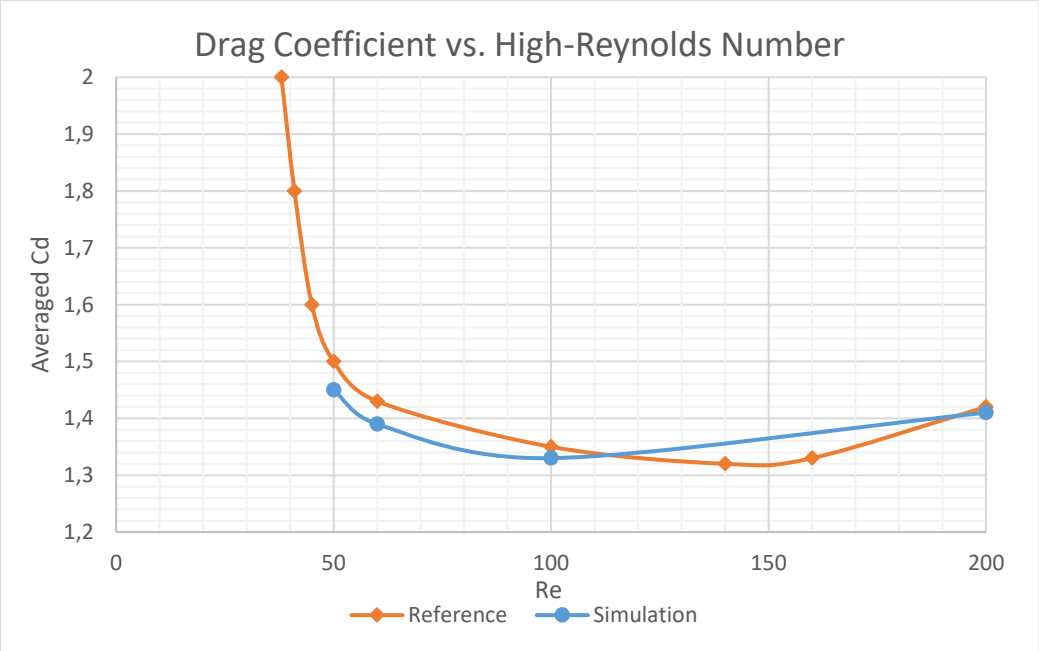


Figure 52. Square Cylinder: Cd vs. Re, a) Steady, and b) Non-steady states

It must be clarified that for low-Re (cases equal to 1, 10, 50, and 60) a steady condition is possible to achieve. Therefore, the values in *Figure 51* are the values at the steady condition. On the other hand, for high-Re (equal to 100 and 200) the steady condition is not reachable. Therefore, the mean value of Cd and Cl was used as shown in *Figure 52*. The values and graphics were presented in order to compare the results with the reference data in [13].

3.2.2.6. Discussion of the Results

First of all, it could be concluded that the results obtained by the algorithms developed *Figure 49, and 52* are highly similar to the reference [13].

- It could be seen in *Figure 49* the flow at low-Re and how it becomes turbulent whilst the Reynolds is increasing. For low-Re the steady condition is reached up until Re = 60 (taking twice computational time) for the algorithm used. Nonetheless, according to the reference, the critical transition value would be around Re = 54, however, because of mesh size and numerical scheme differences it may vary. For high-Re the steady state is not reachable, so the values and results become more instable. In *Figure 52* could be seen how the results are somewhat different from the reference, but they are still highly accurate.
- *Figure 50* shows the velocity field isolines, and it could be concluded that for low-Re there is an attachment of the fluid to the surface of the obstacle, in this case the square cylinder, behind the leading surface. For Reynolds RE = 1, and 10 the isolines are attached to the entire surface and straight to the main flow. As Reynolds increases, the attachment is less notorious, for Re = 50 and 60, it could be seen the creation of recirculating vortexes. For High-RE, the critical Reynolds is reached (RE = 100 and 200) and the flow starts fluctuating; thus, there is detachment of the fluid from the surface, and the vortexes describe more turbulent flows. For RE = 200 it could be seen how the detachment occurs in the bottom and top surfaces of the square cylinder too.
- Considering the aerodynamic analysis, the results were as expected as well. The lift coefficient was expected to be close to zero because of the geometry of the square cylinder. It could be seen in *Figure 51* the variance of Cl for Re = 1 where the mean value, and the steady value is close to zero. Furthermore, the Drag coefficient does have a similar behavior as seen in *Figure 51* where the variance is bigger in the first iterations and whilst it converges the value of Cd becomes stable.
- Regarding the stability of the algorithms utilized, for low-Re values it was necessary using Upwind scheme. QUICK scheme was used for high-Re because it performed better and was more stable.
- Besides the numerical scheme, the CFL condition was changed and the diffusive and convective coefficients were decrease in a factor of 0.6 in order to reach convergence of the code. This effect was mainly noted when using Upwind scheme, and it could be caused by an insufficient mesh refinement.

3.3. Chapter Summary

The Navier-Stokes equations were numerically solved in this chapter by using FSM and furtherly applied to two problems. FSM is one of the most common and easiest method to get a numerical resolution of the N-S equations. Nonetheless, other computational parameters were analyzed during the study, such as mesh sizes, numerical schemes, Reynolds number, and CFL conditions. In the last study an aerodynamic analysis was included.

First, the mesh size was studied in the first problem, Driven Cavity. Results shows how coarsest meshes increased the diffusive effect in the solution. This will affect the selection of the correct numerical scheme and affect the computational time taken by the convergence criteria. It could be concluded as well that the mesh refinement must be carefully chosen because the computational cost could be highly incremented without gaining a significant deal of accuracy.

For the numerical schemes, low order scheme solutions were not accurate enough except for low Reynolds number. This is due t the fact that for low Peclet number, or low convective effects, the upwind scheme is highly accurate and took less computational costs. In general, all the high order schemes give an excellent performance and relatively high computational costs.

Finally, CFL conditions was necessary to be modified to converge the solutions. This parameter had to be reduced by factor of 0.2 until the solution was stable enough. This was necessary for finer meshes, high order schemes, and high Reynolds number.

In the aerodynamic analysis, the lift and drag coefficients were studied during the simulation. the aerodynamic analysis was done for different Reynolds numbers and using a high order scheme.

Chapter 4

INTRODUCTION TO TURBULENCE

Turbulence, or a turbulent flow is characterized by random and chaotic states of motion which causes instabilities in velocity, pressure of temperature fields within the fluid. A turbulent flow could be determined using a dimensionless parameter: the Reynolds Number *Eq. 105*. Re (simplified) is a scalar that correlates de dynamic and the viscous stresses in a fluid. A flow above the critical Reynolds number is considered a turbulent flow, and below it, it becomes a laminar flow.

Turbulence creates rotational fluctuations or so-called turbulent eddies. These rotational structures are generated in a wide range of length scales. According to Kolmogorov K41[17] the minimum time-step and spatial discretization necessary to solve all the scales in a turbulent flow depends on the Reynolds number:

$$\delta t \sim \mathrm{Re}^{-1/2} \qquad \textit{(Eq. 136a)}$$

$$\delta t \sim \mathrm{Re}^{-3/4} \qquad \textit{(Eq. 136b)}$$

Regarding these parameters, and assuming an ideal algorithm using a Direct Numerical Simulation (DNS), the memory requirements grow to approximately $\mathrm{Re}^{9/4}$, and the computational cost to $\mathrm{Re}^{11/4}$[17]. That is the reason why a DNS using the non-linear model from the Navier-Stokes equations is not computationally efficient.

Alternatively, the Burgers equation Eq. have been highly studied because its simpler model and high accuracy that have similarities with the NS equations.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{1}{Re}\frac{\partial^2 u}{\partial x^2} + f \qquad \textit{(Eq. 137)}$$

It should be noted that Burgers equation *Eq. 137* is defined uniquely in 1D space.

In this chapter, the Burgers equation will be defined in the Fourier space, and furtherly solved using DNS and LES simulations. These methods will be applied to a particular case and finally draw some conclusions.

4.1. Burgers Equation in Fourier Space

Considering *Eq. 137* on an interval Ω which is under periodic boundary conditions, it is possible to describe Burgers equation in Fourier space:

$$\frac{\partial \hat{u}_k}{\partial t} + \sum_{k=p+q} \hat{u}_p i q \hat{u}_q = -\frac{k^2}{Re}\hat{u}_k + F_k \quad k = 0, \dots, N \qquad \textit{(Eq. 138)}$$

Where F_k is the forcing term given by $F_k = 0$ for $k > 1$ and F_1 such that $\frac{\partial \hat{u}_1}{\partial t} = 0$ for $t > 0$.

The *k*-th Fourier coefficient of $u(x,t)$ is defined by:

$$u(x) = \sum_{k=-N}^{k=+N} \hat{u}_k e^{ikx} \quad where \quad \hat{u}_k(t) \in \mathbb{C} \qquad \textit{(Eq. 139)}$$

N is the total number of Fourier modes.

It is important to note that $u(x,t) \in \mathbb{R}$; therefore, the *k*-th Fourier coefficient must be equal to its complex conjugate *Eq. 140*.

$$\hat{u}_k = \overline{\hat{u}_{-k}} \qquad \textit{(Eq. 140)}$$

Consequently, the energy E_k of the *k*-th Fourier mode could be obtained with the product of:

$$E_k = \hat{u}_k \cdot \overline{\hat{u}_k} \qquad \textit{(Eq. 141a)}$$

$$\frac{\partial E_k}{\partial t} = -\frac{2k^2}{Re}E_k - \left(\overline{\hat{u}_k} C_k(\hat{u}_p, \hat{u}_q) + \overline{C_k(\hat{u}_p, \hat{u}_q)}\hat{u}_k\right) + \overline{\hat{u}_k} F_k + \overline{F_k}\hat{u}_k \qquad \textit{(Eq. 141b)}$$

Where:

$$C_k(\hat{u}_p, \hat{u}_q) = \sum_{k=p+q} \hat{u}_p i q \hat{u}_q \quad ; \quad C_k(\hat{u}_p, \hat{u}_q) \in \mathbb{C} \qquad \textit{(Eq. 142)}$$

Now, it could be concluded that the diffusive term of the Energy E_k which is given by $-\frac{2k^2}{Re}E_k$ is restraining the energy. Furthermore, that the convective term given by $-\left(\overline{\hat{u}_k} C_k(\hat{u}_p, \hat{u}_q) + \overline{C_k(\hat{u}_p, \hat{u}_q)}\hat{u}_k\right)$ is transporting the energy from large scales to small scales and vice versa, or so-called backscattering effect.

4.2. Case of Study

4.2.1. Burgers Equation

4.2.1.1. Objective

The process to analyze the Burgers equation can now be applied to verify the effectivity of the algorithm. The problem is proposed by CTTC, and it consist of the numerical resolution of the Burgers equation for a fixed Reynolds number, implementing DNS and LES.

4.2.1.2. Problem Definition

The Burgers equation must be solved with $Re = 40$, and taking as initial condition $\hat{u}_k = k^{-1}$

It is assumed that there is no mean flow, therefore $\hat{u}_0 = 0$. To resolve the DNS cases, it will be used $N = 20,\ N = 100$ Fourier modes. Moreover, for the LES will be used $N = 20$ but with two different values of the Kolmogorov constant $C_k = 0.4523, C_k = 0.05$.

Figure 53 shows the expected results given by the CTTC simulation[17] which summarizes the results of the energy spectrum for the steady-state.

The ideal solution will reproduce the results showed in *Figure 53*, and further it could be done a deeper study of the effects of the Reynolds number and number of Fourier modes.

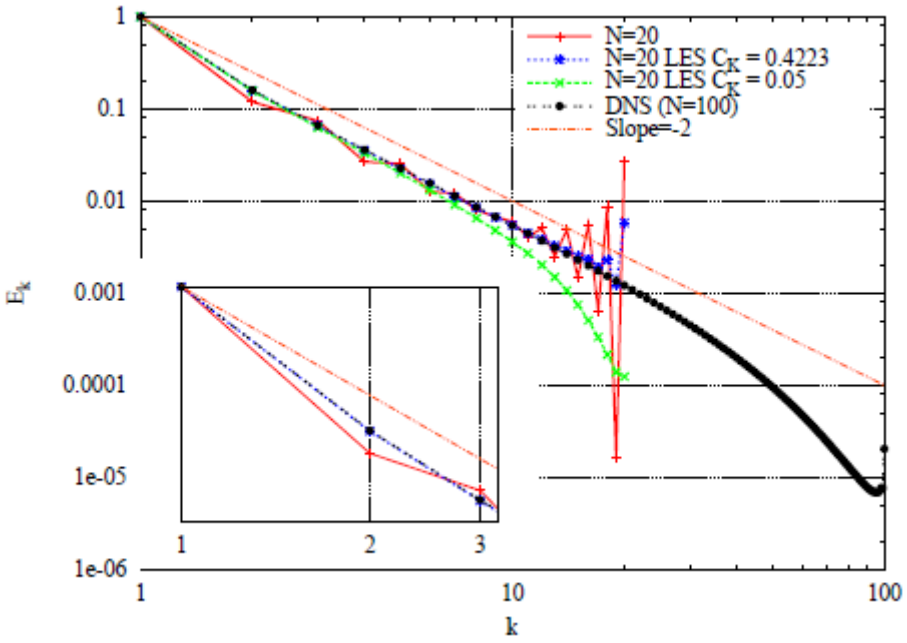


***Figure 53. Burgers Equation: Reference results of the energy spectrum of the steady-state. Extracted from* [17]**

4.2.1.3. Large-Eddy Simulation (LES)

LES offers a great computational advantage for turbulence flow simulations. While DNS would require finer meshes and more Fourier modes, LES can give accurate results with coarser meshes and with less Fourier modes, resulting in a considerable reduction of the computational costs.

The Smagorinsky model[18] was proposed in the mid-60s and is highly implemented because is the simplest LES model[17]. Nonetheless, this model cannot be implemented in Fourier space, therefore, it was proposed to implement a spectral eddy-viscosity model, initially the k-dependent eddy viscosity introduced by Kraichnan[19] with an energy spectrum slope of *-5/3*, and later an improvement for different slopes was proposed by Metais and Lesieur[20]:

$$\nu_t(k/k_N) = \nu_t^{+\infty}\left(\frac{E_{kN}}{k_N}\right)^{1/2}\nu_t^*\left(\frac{k}{k_N}\right) \qquad \textit{(Eq. 143)}$$

Where:

$$\nu_t^{+\infty} = 0.31\frac{5-m}{m+1}\sqrt{3-m}C_k^{-3/2} \qquad \textit{(Eq. 144)}$$

With m as the energy spectrum slope, E_{kN} is the energy at the cut-off frequency k_N, and C_k is the Kolmogorov constant. ν_t^* *Eq. 145* is a non-dimensional eddy-viscosity which is equal to 1 for small values of k/k_N.

$$\nu_t^*\left(\frac{k}{k_N}\right) = 1 + 34.5e^{-3.03(k_N/k)} \qquad \textit{(Eq. 145)}$$

For the 1D case of Burgers equation the slope of the energy spectrum is approximately equal to $m=2$. Finally, the final viscosity in function of the Fourier modes *Eq. 147*, is redefined including the extra viscosity that LES proposes:

$$\nu_{eff}(k) = \nu + \nu_t(k) \qquad \textit{(Eq. 146)}$$

$$\nu_{eff}(k) = \frac{1}{Re} + \nu_t^{+\infty}\left(\frac{E_{kN}}{k_N}\right)^{1/2}\nu_t^*\left(\frac{k}{k_N}\right) \qquad \textit{(Eq. 147)}$$

Finally, using a fully explicit time-integration scheme, it is necessary to define a CFL-like condition to guarantee the code convergence[17]:

$$\Delta t < C_1\frac{Re}{N^2} \qquad \textit{(Eq. 148)}$$

4.2.1.4. Algorithm

With the purpose of graphically explain the code done to solve the Burgers equation, *Figure 54* shows the algorithm flow chart that briefly explain the main processes inside the code. The code consists of one main code, which is the one that is used to manually enter parameters and run the code, and two more sub-codes: one for DNS (Direct numerical simulation) and one for LES (Large-Eddy simulation). See *Attachment I – 2.3* for the algorithm implemented.

4.2.1.5. Results

The results proposed by CTTC in [17] are presented in *Figure 55.* As it could be seen, the results were reproduced significantly equal. Furthermore, a deeper study based on the effects of the Reynolds number was performed as well. A lower Reynolds value (Re = 10), and a higher value (Re = 80) were solved by the algorithm and the results are presented in *Figure 56*, and *Figure 57* respectively.

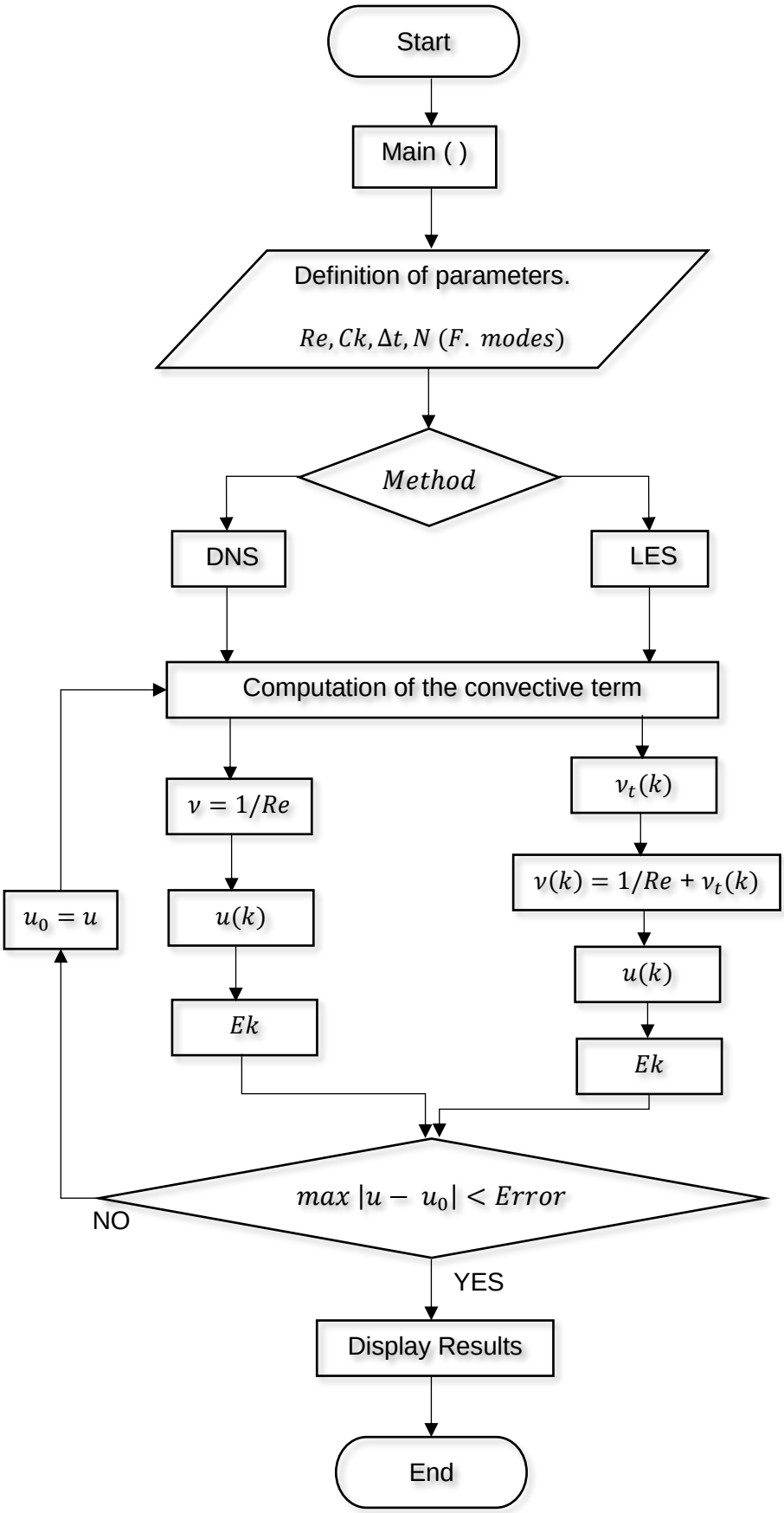


Figure 54. Burgers Equation: Algorithm Flow chart

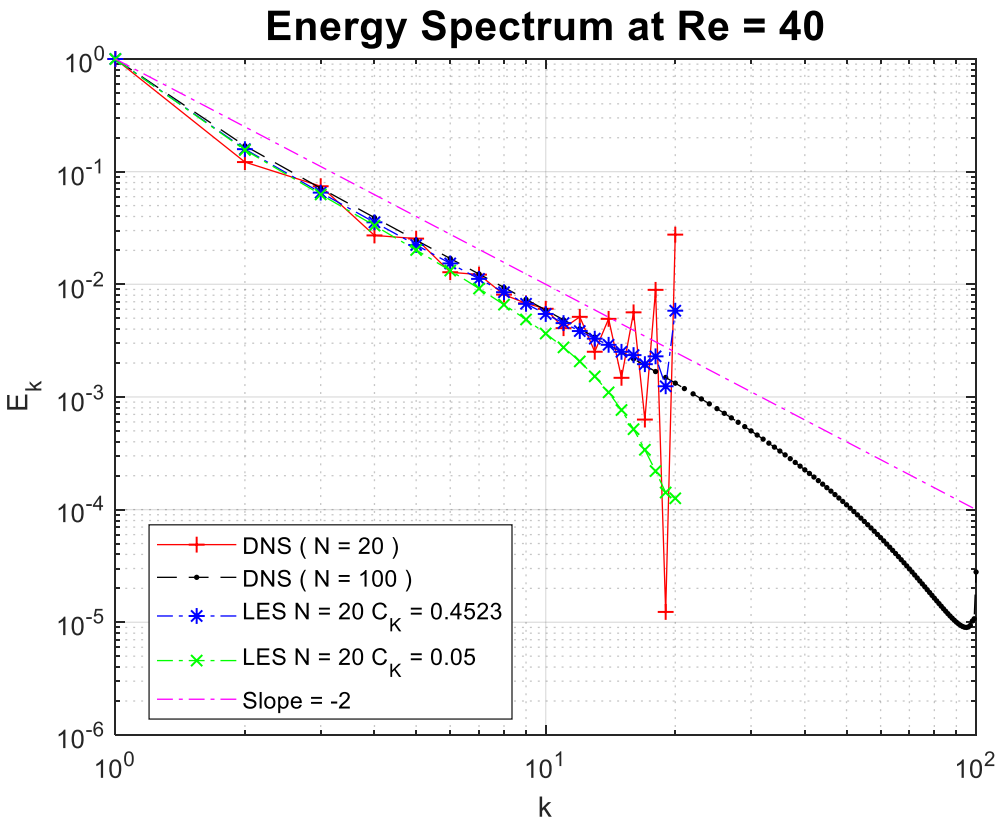


Figure 55. Burgers Equation: Results of simulation DNS, LES at Re = 40

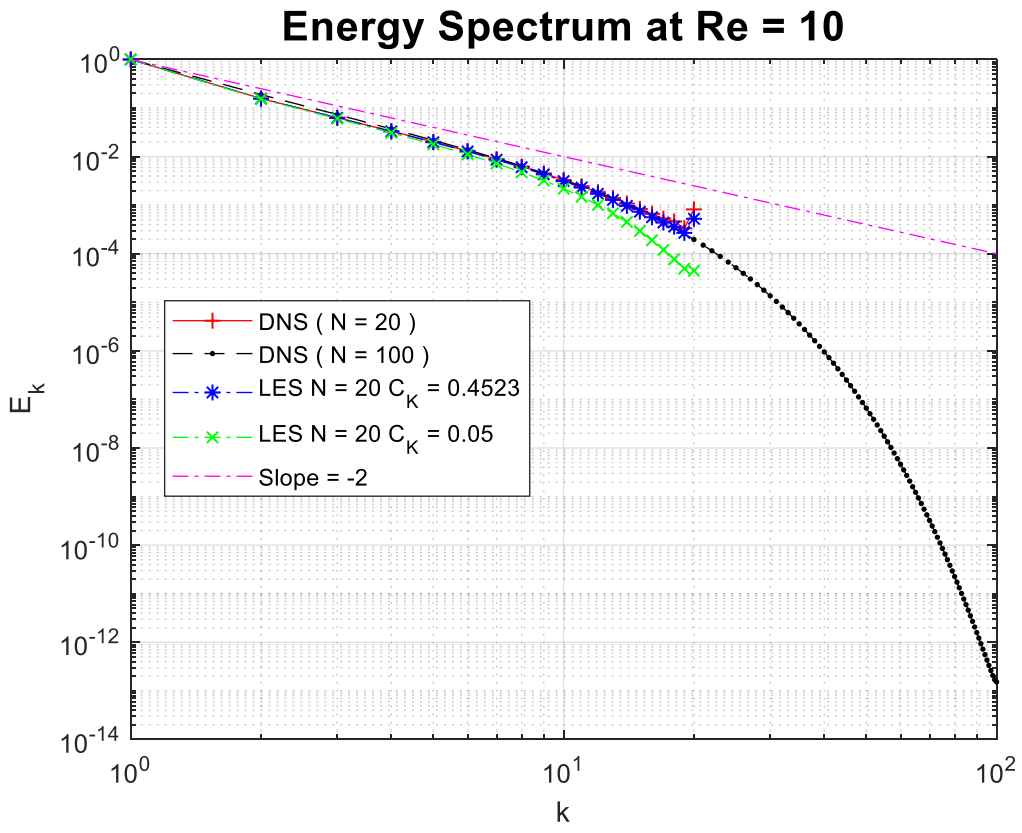


Figure 56. Burgers Equation: Results of extra-simulation DNS, LES at Re = 10

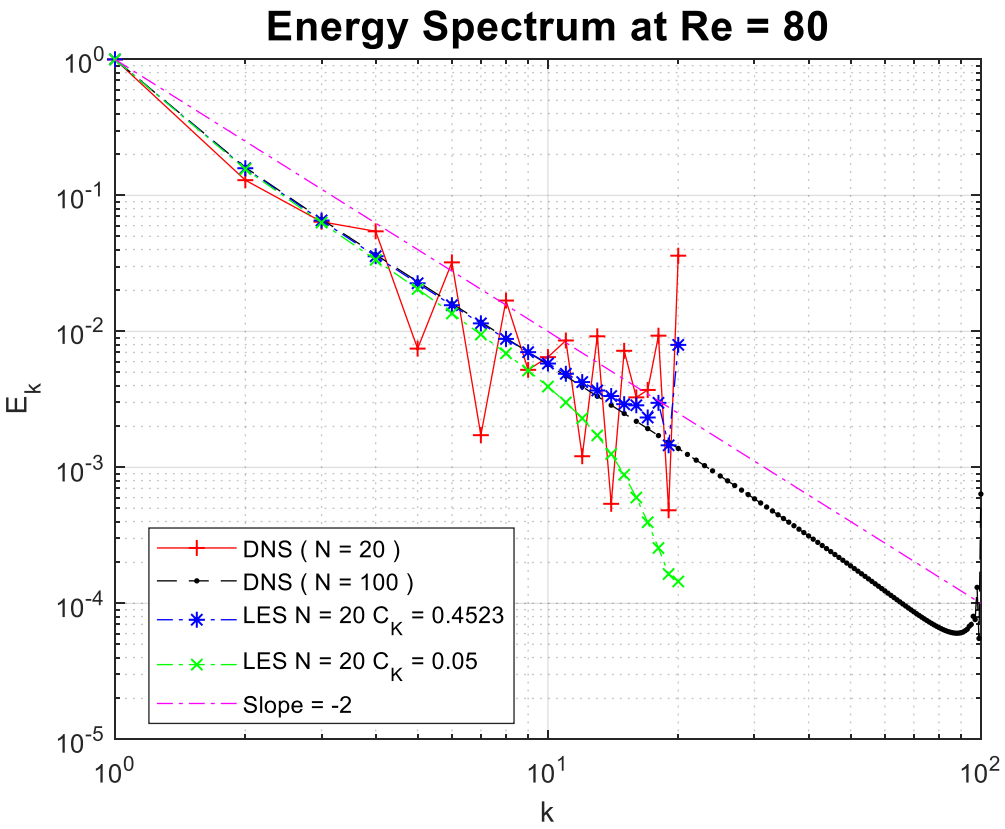


Figure 57. Burgers Equation: Results of extra-simulation DNS, LES at Re = 80

4.2.1.6. Discussion of the Results

First of all, the algorithm implemented for the resolution of the Burger equation gives accurate results and is capable of perform efficiently for DNS and LES methods. It could be seen comparing *Figure 54* with *Figure 52* that both results are highly similar.

- When analyzing DNS could be concluded that it is necessary to use a large number of Fourier modes, as it could be seen in the results that when N=20 the simulation is unstable, and the energy is not well distributed, giving inaccurate results. For N=100 the results are much more accurate, however, the computational l cost is almost 800 % more compare with DNS and LES with N=20.

- On the other hand, LES could perform almost as accurate than DNS with N=100, but just using 20 modes. When using Ck = 0.4523 the results is similar with an exception of the last modes. An interesting fact is that when Ck=0.05 the energy

dissipation is higher, so the energy spectrum decreases more rapidly than with Ck=0,4523.

- It could conclude regarding the variation of the Reynold number, that with low-Re *Figure 55* the diffusive effect is higher, and the energy dissipation is considerable bigger, especially in the last Fourier modes. On the contrary, for higher Reynolds number *Figure 56*, the diffusive effect is reduced, therefore the energy spectrum is closer to the slope (m = -2). It should be noted that ah high Reynolds, DNS with N=20 is totally unstable, and the energy dissipation occurs during all the modes. This is because at higher Reynolds number the influence of the small eddies of turbulence generate more instabilities.

4.3. Chapter Summary

In this chapter, the Burgers equation was discretized and analyzed as a briefly introduction to turbulence. Burgers equation is the simplest way to analyze turbulence. DNS and LES simulation methods were studied in the proposed case. Finally, an extra study with different Reynolds number was done and conclusions were drawn.

First, Burgers equation was discretized in Fourier space, after that, LES simulation was defined to be implemented. The results showed that LES could give an accurate performance using much less computational resources than DNS. DNS can capture all the energy present in large and small eddies, on the contrary, LES uses just the larger eddies which are the ones that carry more energy. Therefore, LES can give an accurate result with a low computational cost.

Chapter **5**

CONCLUSIONS

The *Centre Tecnologic de Transferencia de Calor* CTTC from the Universitat Politecnica de Catalunya proposes this study to engineering students who are interested in computational fluid dynamics CFD. During the study it is possible to get a clearer idea and physical meaning behind the fluid dynamics. Moreover, the development of computational skills required in the modern engineering field.

- The results of this study have shown that the algorithms self-developed were designed and implemented correctly and according to the theory. The algorithms are ready to solve different cases for a wide range of conditions. Furthermore, these algorithms have the ability to adapt to other real cases and give a good performance and accurate results.
- The importance of the convection-diffusion equation when analyzing fluid dynamics has been evaluated. The diffusive and convective effects when apply to a real problem have been analyzed and studied using the algorithms of the cases Diagonal flow, Smith-Hutton, and 2D Heat conduction transfer. Furthermore, the computational implications of using different numerical schemes and numerical solvers which could lead to undesirable effects giving unphysical results.
- One of the main goals was to discretize and numerically solve the Navier-Stokes equations for industrial and academic cases. In that order, the algorithm of Fractional Step Method (FSM) was developed and tested for two definitive cases (Driven Cavity and Flow around Square cylinder).
- The Driven cavity case was analyzed to obtained velocities and pressure fields and compare it to reference data. The algorithm was proven right, and the FSM could be confirmed. This process led to the implementation of the next case.
- Flow around Square Cylinder was the second case of study which involved an aerodynamics study too. Velocity and pressure fields were again obtained to compare it with reference data having an excellent outcome. The aerodynamic analysis was successfully implemented as well.

- The turbulent flows generated by high-Re were analyzed by the last two cases of FSM. Even though the algorithm achieved the goals of the study, high-Re implies turbulent and random flows which is a problem for the computational point of view. That is the reason why the discrepancies that may appear in the results are due to the computational features, such as mesh sizes or numerical scheme, implemented during the simulation of turbulent flows.
- Considering the issues caused by turbulent flows, an introduction to the solution of the Burgers Equation was analyzed too, so in that way a complete study of fluid dynamics will be done and could be integrated to the rest of the work. DNS and LES method to capture the turbulent flow properties were studied giving accurate results and having confirmed the viability of the developed algorithms.
- It could be concluded that CFD is an important field in the modern engineering that involves physical sense, theorical knowledge, and computational skills. This study has proven that the computational resources used could affect the physical results. CFD implies the engineering process as well of choosing the right computational resources and optimization process to obtain accurate and real-like simulations.

5.1. Improvements and Future Work

The study, algorithms, and results achieved the main goals proposed by CTTC as shown before. However, some improvements and deeper studies can be done related to this topic that will enhance the quality, efficiency of the codes, or they could give more interesting results. The main improvements and future work can be summarized in the following list:

- Implementation of new computational techniques (e.g. vectorization) for the algorithm could improve the computational cost by reducing running time.
- Deeper analysis of new numerical schemes, and solvers. Even though high and low order numerical schemes were analyzed, the study showed that for some cases it is necessary to choose the correct one. Also, the computational cost of some of them could be related to boundary and initial conditions.
- Enhancement of the aerodynamic analysis. So far lift and drag coefficient were found, however, it could be interested o analyzed other aerodynamic parameters (e.g. Strouhal number).
- Integration of algorithms with the ability to solve high-Re (i.e. turbulent flows) with the appropriate numerical schemes, numerical solvers, and finer meshes.

Chapter 6

TASK PLANNING

The tasks done by this study are described in *Table 33* which summarize the tasks description, precedence, and times. Moreover, *Table 34* shows the Gannt chart accomplished by the end of this study. A proposed schedule for improvements and future work in *Chapter 6.2* is presented as well in *Table 35*.

6.1. Tasks Description

Task	Description	Precedence	Time [h]
A	Evaluation of the state of the art	-	23
B	Development of algorithm for Convection-Diffusion Equation	A	63
C	Running algorithms developed for the Cases of study for Convection-Diffusion and verifying the results (Diagonal Flow, Smith-Hutton, and 2D Heat Transfers)	B	88
D	Development of FSM algorithm	A, C	76
E	Running algorithms developed for the Cases of study for FSM and verifying the results (Driven Cavity, and Flow around Square Cylinder)	D, C	109
F	Development of algorithm for Burgers Equation	A	18
G	Running algorithms developed for the Cases of study for Burger Equation and verifying the results (DNS, and LES)	F	2
H	Drawing final conclusions and and inclusion of the results to the final document report	G, E, C	16
I	Construction of the final report	A	130

Table 33. Tasks specifications*

*It must be clarified that the total amount of hours presented in *Table 33* is 525, but it is not the real time invested in the study. Task I (construction of the final report) was performed

during test and running codes (Tasks C and E) see *Table. 34,* Therefore, the total time invested was **395 hours.**

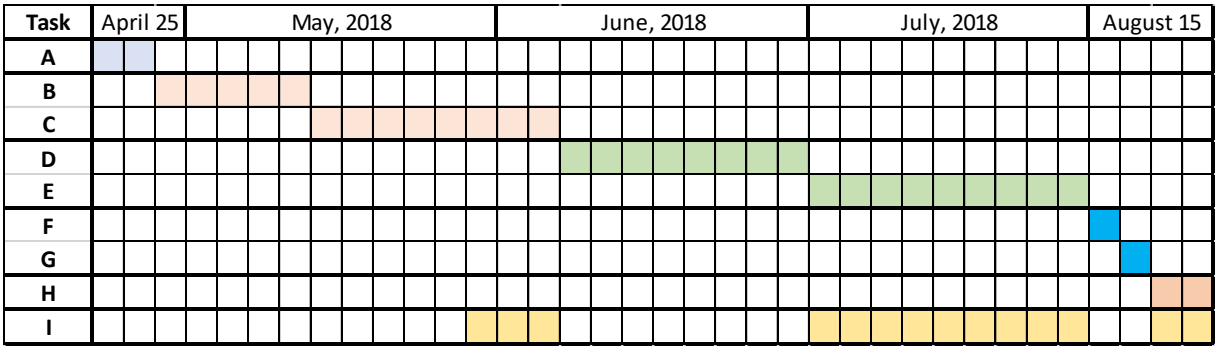


Table 34. Study Gantt chart

6.2. Future Tasks Estimations

Table 35 shows the tasks designation of the future work tasks and its estimated times. This tasks and proposed improvements were deduced of the present study and are conditioned to the probation of the director if this study.

Task	Description	Precedence	Time Estimated [h]
A	Implementation of new computational techniques for all the algorithm developed	-	70
B	Deeper study of numerical schemes and solvers.	A	90
C	Enhancement of the aerodynamic analysis	A	40
D	Integration of algorithms, finer meshes, efficient solvers and numerical schemes to solve turbulent flows	B, C	180

Table 35. Future work tasks and estimated times

ENVIRONMENTAL IMPACT

The environmental impact of this study would become hard to identify for two main reasons: first, there is no final product or physical element; thus, there is no contaminant agents thrown to the ambient. Second, because the study was mainly conducted in a computer and the resources were limited to this condition. However, it is possible to analyze the resources used during the study and the possible improvements to minimize the environmental impact of this study.

- **Paper consumption**: Minimum: Paper was mainly used to take notes from lectures, and to print the final document related to this study
- **Power consumption**: Medium: the most resourced used was electrical power for the computer where codes were developed and run. Because of the complexity of the algorithms, the tests could take hours or even days running and consuming a great amount of power.

The following recommendations could lead to a reduction of the environmental impact for future studies or the continuation of the same:

- Use of recycled paper or take advantage of past lectures notes taken by classmates.
- Rational use of the computer for necessary tasks at the time of code-running, so then the computer will use less power and the RAM performance will be better. Ecological mode is advised (if PC has it) in order to reduce power use of screen and RAM high-performance.

REFERENCES

[1] A. Gibiansky, "Fluid Dynamics : The Navier-Stokes Equations," 2011. [Online]. Available: http://andrew.gibiansky.com/blog/physics/fluid-dynamics-the-navier-stokes-equations/. [Accessed: 07-May-2018].

[2] R. N. Pinto, A. Afzal, L. V. D'Souza, Z. Ansari, and A. D. Mohammed Samee, "Computational Fluid Dynamics in Turbomachinery: A Review of State of the Art," *Arch. Comput. Methods Eng.*, vol. 24, no. 3, pp. 467–479, 2017.

[3] C. D. Pérez segarra, "1b-Convection-diffusion equations - CTTC - manuscript," Barcelona, 2014.

[4] S. V. Patankar, *Numerical heat transfer and fluid flow*. 1980.

[5] M. S. Darwish and F. H. Moukalled, "Normalized variable and space formulation methodology for high-resolution schemes," *Numer. Heat Transf. Part B Fundam.*, vol. 26, no. 1, pp. 79–96, 1994.

[6] CTTC - Universitat Politècnica de Catalunya, "Verification strategies for the convection-diffusion equation," Barcelona, 2014.

[7] CTTC - Universitat Politècnica de Catalunya, "A3-A Two-dimensional Steady Convection-Diffusion Equation : the Smith-Hutton problem," Barcelona, 2014.

[8] CTTC - Universitat Politècnica de Catalunya, "A Two-dimensional Transient Conduction Problem," Barcelona, 2014.

[9] NASA, "Navier-Stokes Equations 3 -dimensional - unsteady," 2015. [Online]. Available: https://www.grc.nasa.gov/www/k-12/airplane/nseqs.html. [Accessed: 07-May-2018].

[10] "Navier-Stokes Equations," 2018. [Online]. Available: https://www.comsol.es/multiphysics/navier-stokes-equations. [Accessed: 07-May-2018].

[11] CTTC - Universitat Politècnica de Catalunya, "Fractional Step Method Staggered Meshes," 2014.

[12] U. Ghia, K. N. Ghia, and C. T. Shin, "High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method," *J. Comput. Phys.*, vol. 48, no.

3, pp. 387–411, 1982.

[13] M. Breuer, J. Bernsdorf, T. Zeiser, and F. Durst, "Accurate computations of the laminar flow past a square cylinder based on two different methods: Lattice-Boltzmann and Finite-Volume," *Int. J. Heat Fluid Flow*, vol. 21(2), pp. 186–196, 2000.

[14] NASA, "The Lift Coefficient," 2015. [Online]. Available: https://www.grc.nasa.gov/WWW/K-12/airplane/liftco.html. [Accessed: 08-Jul-2018].

[15] NASA, "The Drag Coefficient," 2015. [Online]. Available: https://www.grc.nasa.gov/www/k-12/airplane/dragco.html. [Accessed: 08-Jul-2018].

[16] J. Katz and A. Plotkin, *Low-Speed Aerodynamics*, Second Edi. New York: CAMBRIDGE UNIVERSITY PRESS, 2001.

[17] CTTC - Universitat Politècnica de Catalunya, "Burgers equation in Fourier space," pp. 1–8, 2014.

[18] J. Smagorinsky, "General Circulation Experiments with the Primitive Equations.," *J. Fluid Mech.*, vol. 91, pp. 99–164, 1963.

[19] R. H. Kraichnan, "Eddy viscosity in two and three dimensions.," *J. Atmos. Sci.*, vol. 33, no. 8, pp. 1521–1536, 1976.

[20] M. Lesieur and O. Metais, "Spectral large-eddy simulation of isotropic and stably stratified turbulence.," *J. Fluid Mech.*, vol. 239, pp. 157–194, 1992.